Tensor Methods for Signal Reconstruction and Network Embedding

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Charilaos I. Kanatsoulis

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Prof. Nikolaos D. Sidiropoulos, Advisor

October, 2020

Acknowledgments

First, I would like to express my sincerest gratitude to my advisor, Prof. Nikolaos D. Sidiropoulos, for welcoming me as a member in his cream of the crop academic group and giving me the opportunity be a student at the University of Minnesota. His guidance, invaluable advice and dedication to research, teaching and ethics have always served as an inspiration and helped me excel not only as a researcher but also as a person. It has been a pleasure and a privilege being his student.

I would also like to extend my deepest thanks to Prof. Georgios Giannakis, Prof. George Karypis, Prof. Mingyi Hong, and Prof. Mehmet Akçakaya for serving in my doctoral committe and providing valuable comments and feedback on my research and thesis. I would also like to deeply thank my friend and collaborator Prof. Xiao Fu. His help and mentorship in my first research steps were pivotal and resulted in co-authoring some of my favorite papers. I am also grateful to Prof. Wing-Kin Ma. It was his presentation at the University of Minnesota that gave birth to the tensor sampling idea, which is the core of this thesis.

Moreover, I am indebted to all of my professors at the University of Minnesota. Their effort and teaching provided me with essential tools to conduct my research and develop my ideas. Special thanks go to all the current and previous lab-mates: Dr. Balasubramanian Gopalakrishnan, Prof. Kejun Huang, John Tranter, Dr. Aritra Konar, Dr. Cheng Qian, Dr. Bo Yang, Dr. Ahmed Zamzam, Nikos Kargas, Faisal Almutairi, Dr. Panagiotis Alevizos, Dr. Yunmei Shi, Dr. Tianyu Qiu, Dr. Mikael Sorensen, Magda Amiridi, Mohamed Salah and Paris Karakassis

I am forever grateful for all the friends I made in Minnesota: Panos, Dimitris, Diamantis, Thanasis, Yiannis, Fatemeh, Kostas, Lambros, Dennis, Spyros, Donghoon, Thanos, Maria, Nikos, Vassilis, Michael, Vassilis, Vassilis, Vassilis, Andreas, Agi, Ioanna, Eva, Konstantina, Dimitris, Argyris, Yiota. This section would not be complete without thanking all my lifelong friends in Greece.

Last but not least l would like thank my family and especially my parents, Yianni and Katerina for being my role models and my sister Maro for being the best twin someone can ask for.

Charilaos I. Kanatsoulis, Minneapolis, October, 2020.

Dedication

This dissertation is dedicated to my family.

Abstract

Over the past few years, the avalanche of data along with advances in methodological and algorithmic design have triggered an increased interest in *machine learning* (ML) and *signal processing* (SP) research. How do we fuse and complete multi-dimensional signals? What is a concise and informative representation of entities in multi-dimensional networks? How do we develop efficient lightweight algorithms that handle very large data? These are important questions that have risen on the top of the scientific and engineering agenda of ML and SP communities. A plethora of methods has been proposed to answer such questions. While neural networks are the current trend and powerful non-linear data-driven tools, there exist principled alternatives, such as multi-linear tensor methods, that are also effective and oftentimes significantly outperform neural network approaches.

In the era of data deluge, multi-dimensional data, also known as *tensors*, are ubiquitous in a number of engineering tasks and data analytics. Tensors can model various types of data in high-impact domains. Images, for example, are space-space-spectrum cubes that can be naturally represented as tensors. Different types of networks as knowledge graphs and networks with attributed nodes are also tailored to tensor modeling. On the other hand, *tensor decompositions* have proven essential tools in understanding, analyzing and processing multi-dimensional data. They offer a flexible analytical framework with solid foundations, as well as efficient algorithms that effectively handle multi-dimensional data.

This thesis aims to answer the aforementioned questions by exploiting tensor modeling and decomposition tools. The objective is to propose elegant and effective solutions to a number of challenging machine learning and signal processing problems. In particular three main research thrusts are investigated: i) Hyperspectral super-resolution; ii) Tensor sampling and reconstruction; and iii) Network representation learning. For each of the thrusts, this thesis offers an efficient framework that is supported by theoretical analysis, algorithmic foundations and thorough experimental investigation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Machine learning (ML), *data science* (DS) and *signal processing* (SP) are engineering fields that have gained significant interest over the past decade. Universities, for instance, are adopting ML techniques to recommend courses that students are likely to succeed. Along the same lines, service companies have effective algorithms to recommend movies, music and products that their customers will enjoy. Social networking companies, on the other hand, use ML to predict potential connections between the members of the network. Quantitative analysis and financial forecasting is also of great interest to the vast majority of companies, which employ state-of-the-art DS and ML approaches to handle them. ML and SP have also intruded less traditional fields as precision agriculture and health sciences. In precision agriculture, for instance, ML is applied to detect diseases at a whole plant level before we can visually see them. Accelerating the magnetic resonance imaging scan acquisition process is also an important task in medical imaging that heavily uses SP techniques. The recent spread of COVID-19 pandemic has also motivated scientists to innovate with health data. In particular, ML techniques are employed along the side of medical research for drug repurposing and disease analysis.

Two key factors, among others, were critical in the upsurge of ML, DS and SP techniques. The first is the amount of data that is available for a plethora of science and engineering tasks. Advances in data integration and acquisition have prompted an unprecedented data avalanche. However, the sheer dimension and volume of data is not always a blessing. Real data are usually incomplete, heterogeneous, multi-modal and multi-view. Unifying, completing and processing them can be a challenge. Triggered by these challenges, methods and algorithms have also drastically evolved to meet the new standards. Neural networks is the current trend and also

very effective non-linear tools for a variety of applications. Neural networks, however, are not a panacea. There exist several other ML, DS and SP tools that work very efficiently in a number of applications. For instance matrix factorization, tensor analysis, canonical correlation analysis, random walks, etc., are principled alternatives with solid foundations, that oftentimes are more suitable for certain tasks and outperform neural network approaches.

In the era of data deluge, multi-view, multi-dimensional data and signals are ubiquitous in numerous domains. Modeling them as *tensors*, i.e., multi-way arrays, allows for leveraging *tensor decomposition* techniques, which are powerful multi-linear analytical and processing tools. Tensors and tensor decompositions find applications in various fields including signal processing [55], machine learning [10, 90, 126, 160], data mining [13, 125], remote sensing [80, 84, 84], medical imaging [39, 88], [144], genomics [70], and chemometrics [148], just to name a few. The ability of tensor decomposition and in particular the *canonical polyadic decomposition* (CPD) to capture high-order dependencies across the tensor dimensions, along with their uniqueness and parsimony properties, offer an effective framework to handle these tasks.

This thesis aspires to provide elegant and effective solutions to challenging ML, DS and SP problems. The considered applications involve multi-dimensional data (signals), e.g., images and graphs and tensor decomposition techniques are employed to offer efficient solutions. The proposed framework for each task, combines concise modeling, analytical foundations and efficient algorithmic development. In particular the thesis will develop along three main research thrusts:

- Hyperspectral super-resolution.
- Tensor sampling and completion.
- Network representation learning.

1.1 Hyperspectral super-resolution

Image fusion from multiple sensors has attracted much attention from several communities (e.g., signal and image processing, remote sensing, and computer vision), since it proves very useful in a lot of applications [103, 130, 171]. Recently, the remote sensing community has invested significant effort in fusing hyperspectral and multispectral images– see Fig. 4.7. This technique is known as *hyperspectral super-resolution* (HSR) or *hyperspectral-multispectral*

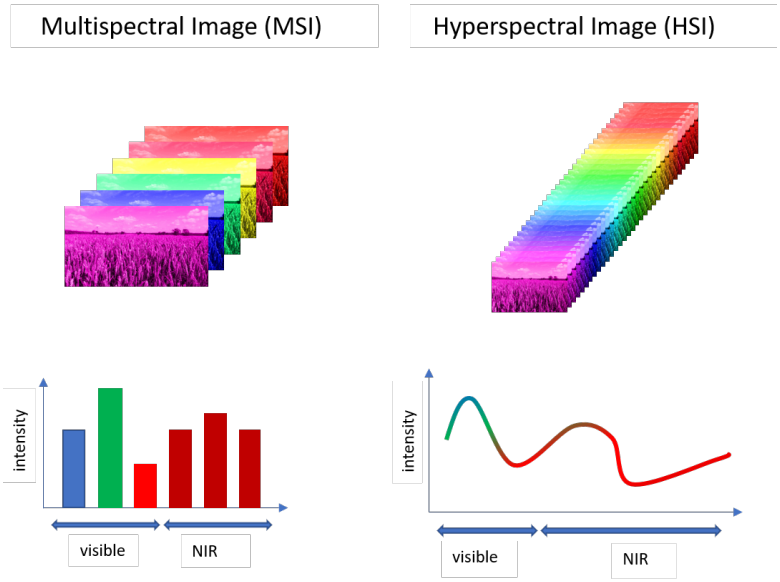


Figure 1.1: Example of an HSI and an MSI.

fusion [185]. The ultimate goal of HSR is to integrate information from a hyperspectral image (HSI), which admits high spectral resolution but coarse spatial resolution, and a (co-registered) multispectral image (MSI), which has fine spatial resolution but low spectral resolution, to produce a super-resolution image (SRI) that admits both high spatial and spectral resolutions. This task is very well-motivated, since an SRI is of great interest to multiple analytical tasks (e.g., small object tracking and identification). However, it is considered very costly to simultaneously improve both the spectral and spatial resolutions of the multiband sensors due to hardware limitations [185]. Nevertheless, HSR techniques allow the construction of an SRI via fusing images that are captured by existing sensors [6, 141].

Chapter 3, presents two novel hyperspectral super-resolution approaches. Our approaches start with the fact that both HSI and MSI images are space-space-spectrum "cubes", and thus can be naturally represented as third-order tensors [143]. Tensors admit a number of favorable properties that matrices do not have. For example, any tensor admit a canonical polyadic decomposition (CPD), which captures dependencies across the different dimensions (or modes)—and this decomposition is essentially unique under mild conditions. The proposed methods employ a coupled CPD model to tackle the HSI-MSI fusion task. We show that both models guarantee the identifiability of the SRI under realistic conditions and the idea is to leverage the

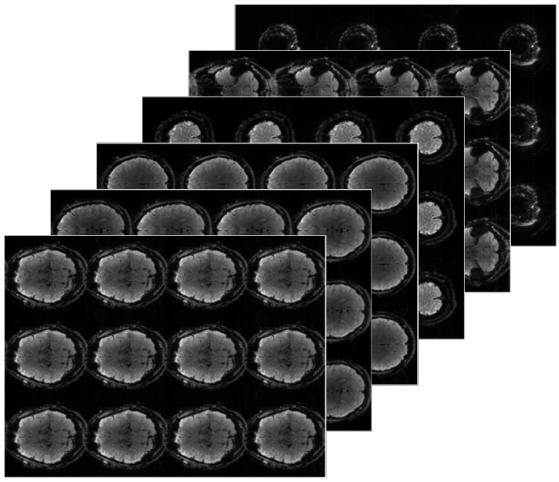


Figure 1.2: Example of an fMRI scan.

uniqueness of the CPD model. Note that identifiability-guaranteed models and algorithms are not only of theoretical interest—they usually offer more favorable empirical results, e.g., exhibiting enhanced-performance and being less sensitive to initializations. Furthermore, the proposed approaches can work under scenarios where the spatial degradation operator is unknown. Unlike some existing methods which attempt to estimate the spatial degradation operator [146, 186], our methods work under the case where the spatial degradation operator is not known at all—without losing identifiability of the SRI. Numerical experiments using synthetic and semi-real data show that the proposed approaches are very promising for the hyperspectral super-resolution task.

1.2 Tensor sampling and completion

Signal sampling and reconstruction is a fundamental engineering task at the heart of signal processing. In the first half of the 20th century, Whittaker, Nyquist, Kotelnikov, and Shannon [96, 120, 139, 178] laid the foundation of the sampling theorem, which together with the discovery of the fast Fourier transform catalyzed the field of signal processing. In order to perfectly reconstruct a signal from uniformly spaced samples, one must sample at a rate at least twice the maximum frequency present in the signal. Unfortunately a large number of signals of interest are far from being band-limited.

Compressive sensing (CS) [31, 32, 49] emerged in the early 2000's as an alternative which allows recovery from a set of measurements sampled or compressed below the Nyquist rate.

CS relies on two basic principles: the signal of interest must be sparse in some domain and the sampling/compression pattern should be 'incoherent'. Compared to the sampling theorem, CS exploits sparsity (instead of bandlimitedness) in a known domain, thereby enabling reconstruction from fewer measurements. However, uniform or regular sampling is more appealing in practice and from the system design point of view, as it is far simpler to implement, and often necessary due to system constraints. The principles of CS, have been extended to multi-dimensional signal as matrices and tensors. The problem is known as low rank matrix completion (LRMC) and low rank tensor completion. The sample complexity is determined by the signal rank, and incoherent sampling patterns are again employed.

Chapter 4 is motivated by the following question. *Is there a sub-Nyquist sampling mechanism that works under regular sampling for certain signals of interest*? This research question is very intriguing: regular sampling is efficient, friendly to implementation and often mandatory, and sub-Nyquist sampling is desired since numerous real-world signals are far from being bandlimitted.

We offer an affirmative answer to the above research question for a large variety of multidimensional signals. We propose a tensor sampling framework that is flexible and easy to implement. Generic as well as deterministic theoretical conditions are derived, under which identifiability is guaranteed. Similar to matrix completion, the sample complexity for tensor signal reconstruction is mainly affected by the tensor rank and the tensor size—instead of signal bandwidth or sparsity. Unlike CS and LRMC, the proposed approach does not require incoherent sampling. Therefore, regular, equispaced and highly structured sampling strategies can be adopted—which has a much broader spectrum of applications in practice.

Our second major contribution lies in designing accelerated acquisition schemes for *functional magnetic resonance imaging* (fMRI) (see Fig. 4.5) utilizing the proposed tensor sampling principles. Note that traditional fMRI acquisition is considered an "agonizingly slow" scanning process, which strongly motivates exploring appropriate sampling techniques for acceleration. However, due to hardware limitations, random or incoherent sampling strategies are considered impractical for this task [52]. Nevertheless, the proposed tensor sampling framework fits this task very well as fMRI signals are naturally tensors. Extensive simulations using synthetically generated data show that the proposed tensor sampling schemes are promising. More importantly, experiments using real fMRI data demonstrate remarkable acceleration compared to traditional fMRI scanning approaches, without sacrificing reconstruction accuracy.

We also use regular tensor sampling mechanisms to design efficient algorithms to compute the CPD for large-scale tensors. Tensors with millions or billions of entries are common in numerous fields. A raw fMRI scan, for instance, can be represented as a dense complex tensor with dimensions $10,000 \times 500 \times 2,000$ which corresponds to 10 billion non-zero complex entries. The NELL dataset [35], which represents real world knowledge base data, is a $26 \times 26 \times 48$ million tensor with 144 million non-zero entries. Standard CPD methods, which are computationally intensive and memory demanding, have difficulty in operating with big data tensors. For an $I \times J \times K$ tensor of rank F each iteration of the popular alternating least squares (ALS) method requires IJF additional memory and $IJF + IJKF$ flops for dense or $IJF + 2Fm$ for sparse tensors, where m is the number of non-zero entries. It is therefore clear that computing the CPD of large scale tensors is challenging.

The simplest idea to overcome these limitation is to use fewer data, when computing the CPD. However, life is not as easy and a naive random sampling of the tensor is likely to fail. The reason is twofold. First, the solution of the computationally lighter problem is not guaranteed to be the same as the solution of the original one. Second, computing the CPD of an incomplete tensor is usually a much more difficult problem compared to the CPD of the full tensor. As a result, even when identifiability is guaranteed the algorithm of incomplete tensor might produce uninteresting results.

Chapter 5 addresses the aforementioned challenges and proposes a regular tensor sampling framework to compute the CPD of large-scale tensors. Specifically, two new multi-modal regular sampling mechanisms are proposed, which are identifiable, i.e., an optimal solution is guaranteed to provide the true factors, and accomplish significant speed-up. Furthermore, a lightweight algorithm is developed to perform the CPD computation and verify its effectiveness via synthetic and real data simulations.

1.3 Network representation learning

Network science studies the behavior of entities, belonging to one or more communities, via observing their mutual interactions [17]. Networks and network science have attracted considerable attention in science and engineering, since they offer an elegant abstraction of various physical, social, and engineered systems – and effective tools to analyze them [50, 118]. Networks are nowadays ubiquitous in a plethora of science and engineering disciplines, including social,

communication, and biological networks, to name a few.

Networks are usually represented by graphs, which are informative abstractions and model the interactions in the system. In particular, graph representations encode the connectivity information of different entities (nodes) through a set of edges. The connectivity information in a network is important and describes each node in the network in terms of the rest of the nodes.

1.3.1 Node embedding of attributed Graphs

In real world networks, the entities are not only defined by their connectivity with other entities, but can also be described by a set of measurements or attributes, which offer a node characterisation at an individual level, and are usually very informative. Although graphs offer an elegant and essential way to represent the entities of a network, it is often the case that an individual representation of an entity is required that is not necessarily described by relations with respect to subsets of the community. Furthermore, when attributes are also available for each node, which is often the case in practice, it is convenient to combine both connectivity and attribute information in a single, universal representation of that node, one that encapsulates as much information as possible. Moreover, a variety of networks of interest involve millions of nodes, which makes graph representation of nodes highly impractical for certain tasks.

The aforementioned challenges underscore the need for concise and informative representation of network nodes that is conducive for exploratory analysis as well as downstream applications. This has motivated a considerable body of research on embedding graph nodes in a low-dimensional vector space, using graph and attribute information in an unsupervised manner. The task is also known as unsupervised node or graph representation learning. The objective of unsupervised node embedding is twofold. On the one hand, the embeddings should capture the maximum amount of knowledge present in the graph and attributes so that information loss is avoided. Towards this end, a key to successful node embeddings is to be able to preserve the geometry of the network, defined by proximity in both the connectivity and the attributes of the nodes. On the other hand the embedding should be able to boost the performance of various downstream network tasks, such as node classification, link prediction, and community detection, to name a few. Concise node representations produced by embedding algorithms can significantly benefit feature-based tools such as logistic regression, support vector machines, and even neural networks – especially when we only have access to limited training data.

The work in chapter 6 is motivated by the following question: *Can we produce node network*

embeddings such that we provably preserve the geometry of 1) the distances associated with the connectivity information of the network, and 2) the distances associated with the attributed information of the network, in an unsupervised manner? This is a well motivated problem, since maintaining the network geometry is a fundamental objective of representation learning, and doing so significantly improves the performance of several downstream tasks.

Chapter 6 introduces Geometry-preserving Attributed Graph Embedding (GAGE) – a principled approach to extract node embeddings in an unsupervised fashion. GAGE enjoys several favorable properties.

- By design, the produced embeddings preserve node geometry, as inferred from both the node adjacency matrix and the node attributes.
- The node embeddings are unique and thus permutation invariant, meaning that any re-ordering of the nodes in the adjacency representation yield the same embeddings.
- The approach is applicable to both undirected and directed networks.
- The proposed approach is flexible and does not require connectivity and attribute information for every node. In other words embeddings can be produced for nodes with partially / completely missing connectivity *or* attribute information (but not both, obviously).
- The proposed algorithm is lightweight and scalable – it can efficiently handle large networks.

The contributions of chapter 6 can be summarized as follows:

- **Novel problem formulation**: Previous work in this area hasn't formalized the intuitive requirement that the embedding should be capable of (approximately) reproducing the distances in terms of connectivity and attribute information.
- **Analysis:** We show that by leveraging the favorable properties of tensor factorization and multi dimensional scaling the proposed embedding can (approximately) reproduce both the connectivity and attribute distances.
- **Algorithm:** We propose a novel tensor factorization algorithm to perform unsupervised embedding task. The algorithm exploits the special structure of the tensor, is fast and scalable for big networks.

- **Experimental verification:** The proposed node embedding approach is assessed under node classification and link prediction settings and exhibits very promising results in both tasks.

1.3.2 Knowldge Graph Embedding

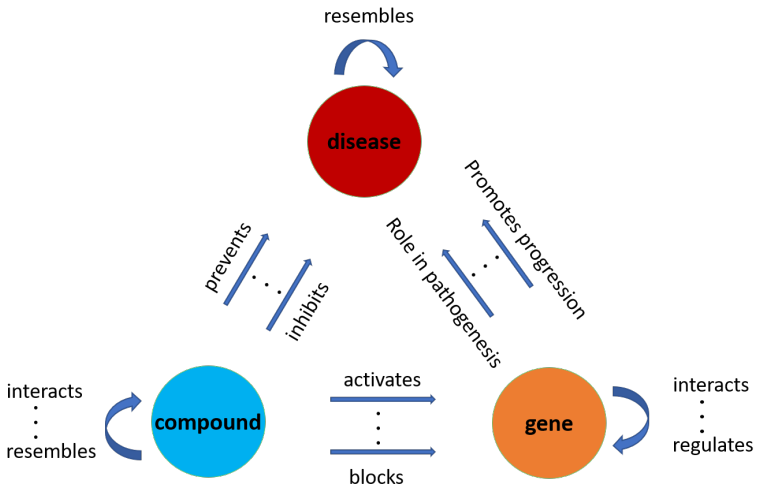


Figure 1.3: Knowledge Graph of biomedical components.

A knowledge graph (KG) is a type of network that models the relational behavior of various entities in knowledge bases. A KG is heterogeneous in the sense that it models interactions between entities of different type, e.g., drugs and diseases, and is also a multidimensional network (edge-labeled multi-graph) [23], since the edges (interactions) that connect the nodes (entities) can be multiple and also of different type. Knowledge graphs (KGs) have recently attracted significant attention due to their applicability to various science and engineering tasks. For instance, popular knowledge graphs are YAGO [155], DBpedia [12], NELL [35], Freebase [27], and the Google KG [147]. A recent trend codifies knowledge bases of biomedical components and processes, such as genes, diseases and drugs into KG's (see Fig. 1.3) e.g., [67, 68, 74]. KGs can model any relations of the form subject-predicate-object, as well as higher-order generalizations. However, this broad modeling freedom can sometimes be a challenge, as the entities can be very diverse and the dimensions of the KG can turn prohibitively large.

Chapter 7 introduces `TeX-Graph`, a novel coupled tensor-matrix framework to perform KG

embedding. The proposed KG coupled tensor-matrix modeling extracts meaningful information from a set of diverse entities with multi-modal interactions in a principled and concise manner. `TeX-Graph` avoids modeling inefficiencies in previously proposed tensor models, and relative to neural network approaches it offers a principled and effective way to produce unique KG representations. The proposed framework is used for drug repurposing, a pivotal tool in the fight against COVID-19 and other diseases. Learning concise representations for drug compounds, diseases, and the relations between them, our approach allows for link prediction between drug compounds and COVID-19 or other diseases. The impact is critical. First, compound repurposing enables drug design that drastically reduces the design exploration cycle and the failure rate. Second, it markedly reduces drug development cost, as developing new therapeutic drugs is tremendously expensive.

The contributions of chapter 7 can be summarized as follows:

- **Novel KG modeling**: We propose a principled coupled tensor-matrix model tailored to KG needs for efficient and parsimonious representations.
- **Analysis:** The `TeX-Graph` embeddings are unique and permutation invariant, a property which is important for consistency and necessary for interpretability.
- **Algorithm:** We design a scalable algorithmic framework with lightweight updates, that can effectively handle very large KGs.
- **Application:** The proposed framework is developed to perform drug repurposing, a pivotal task in the fight against COVID-19.
- **Performance:** `TeX-Graph` achieves 100% performance improvement compared to the best available baseline for COVID-19 drug repurposing using a recently developed COVID-19 KG.

1.4 Thesis Outline

The remainder of the thesis is organized as follows.

Chapter 2 discusses some tensor algebra preliminaries. Two models are introduced, the CPD and the coupled CPD along with conditions for identifiability. The matricization and mode product operation are also defined.

Chapter 3 introduces two novel coupled tensor decomposition approaches to tackle the HSR task. Identifiability guarantees are provided along with an efficient algorithmic framework. Extensive experiments are conducted with real HSI's.

Chapter 4 studies the problem of completing a tensor from regular samples. Three different sampling mechanisms are proposed and conditions are derived under which the the tensor is identifiable. Furthermore, the task of accelerating the fMRI scan acquisition is cast as a regular tensor completion problem and an efficient algorithmic framework is developed to tackle it.

Chapter 5 builds upon the results of chapter 4 and introduces two new regular sampling mechanisms along with efficient algorithms to compute the CPD of large-scale tensors.

Chapter 6 introduces a novel approach to perform node embedding on attributed networks. An efficient algorithm is designed that preserves the connectivity and attribute geometry of the network. The approach is tested on node classification and link prediction tasks.

Chapter 7 studies the problem of KG embedding. A novel coupled tensor matrix approach is proposed and an efficient algorithmic framework is developed. The approach is employed to suggest potential drugs in the fight against COVID-19.

Finally, Chapter 8 presents a concluding discussion of this thesis along with future directions.

1.5 Notational Conventions

The notation used in this thesis is summarized in Table 1.1.

Table 1.1: Overview of notation.

x, y, z	$\triangleq$	scalars
(m, n), (h, r, t)	$\triangleq$	ordered tuple
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	$\triangleq$	vectors
$\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$	$\triangleq$	matrices
$\underline{\boldsymbol{X}}, \underline{\boldsymbol{Y}}, \underline{\boldsymbol{Z}}$	$\triangleq$	tensors
$\mathcal{S}$	$\triangleq$	set
$\boldsymbol{A}(:, f)$	$\triangleq$	f-th column of matrix $\boldsymbol{A}$
$\boldsymbol{A}(i, :)$	$\triangleq$	i-th row of matrix $\boldsymbol{A}$
$\boldsymbol{X}^k$	$\triangleq$	k-th frontal slab of tensor $\underline{\boldsymbol{X}}$
$\boldsymbol{A}^T$	$\triangleq$	transpose of matrix $\boldsymbol{A}$
$\|\boldsymbol{A}\|_F$	$\triangleq$	Frobenius norm of matrix $\boldsymbol{A}$
$\otimes$	$\triangleq$	Kronecker product of two matrices
$\odot$	$\triangleq$	Khatri-Rao (columnwise Kronecker) product
$\circ$	$\triangleq$	outer product
$*$	$\triangleq$	Hadamard product
diag$(\boldsymbol{x})$	$\triangleq$	diagonal matrix of vector $\boldsymbol{x}$
$\lfloor x \rfloor$	$\triangleq$	largest integer that is less than or equal to x
nnz	$\triangleq$	number of non-zeros
$\boldsymbol{I}$	$\triangleq$	Identity matrix
$\mathbf{1}$	$\triangleq$	vector of ones

Chapter 2

Tensor Algebra Preliminaries

2.1 The Canonical Polyadic Decomposition of a tensor

In this thesis we heavily use tensor algebra. To facilitate the upcoming discussion we briefly present some essential tensor algebra concepts. The reader is referred to [93, 143] for further details.

A third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{F}^{I \times J \times K}$ is a three-way array indexed by i, j, k with elements $\underline{\boldsymbol{X}}(i, j, k)$, where $\mathbb{F}$ is used to denote either the real field $\mathbb{R}$ or complex field $\mathbb{C}$. It consists of three modes: columns $\underline{\boldsymbol{X}}(i, :, k)$, rows $\underline{\boldsymbol{X}}(:, j, k)$, fibers $\underline{\boldsymbol{X}}(i, j, :)$; and three types of slabs: horizontal $\underline{\boldsymbol{X}}(i, :, :)$, vertical $\underline{\boldsymbol{X}}(:, j, :)$ and frontal $\underline{\boldsymbol{X}}(:, :, k)$ – see Fig. 2.1, 2.2, respectively.

A rank-one tensor $\underline{\boldsymbol{Z}} \in \mathbb{F}^{I \times J \times K}$ is the outer product of three vectors:

$$\underline{\boldsymbol{Z}}(i, j, k) = \boldsymbol{a}(i)\boldsymbol{b}(j)\boldsymbol{c}(k), \ \ \forall i, j, k, \tag{2.1}$$

where $\boldsymbol{a} \in \mathbb{F}^I, \ \boldsymbol{b} \in \mathbb{F}^J, \ \boldsymbol{c} \in \mathbb{F}^K$. The shorthand notation for the above is $\underline{\boldsymbol{Z}} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}$, where $\circ$ denotes the outer product. Any tensor can be realized as a sum of three way outer products (rank one tensors), i.e.

$$\underline{\boldsymbol{X}} = \sum_{f=1}^{F} \boldsymbol{a}_f \circ \boldsymbol{b}_f \circ \boldsymbol{c}_f. \tag{2.2}$$

The above expression is known as the polyadic decomposition (PD) of a third-order tensor. If F denotes the minimum number of outer products needed to synthesize $\underline{\boldsymbol{X}}$, then F is called *tensor rank* or *CP rank* and the decomposition is known as *canonical polyadic decomposition* (CPD) or

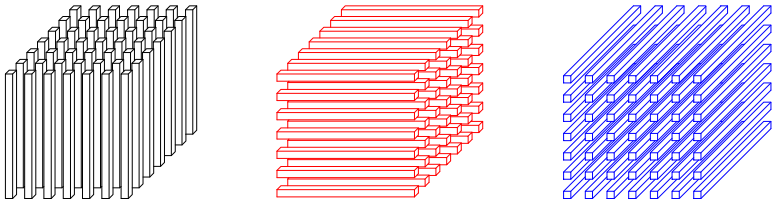


Figure 2.1: The columns ($\underline{\boldsymbol{X}}(i,:,k)$), rows ($\underline{\boldsymbol{X}}(:,j,k)$), and fibers ($\underline{\boldsymbol{X}}(i,j,:)$) of a third-order tensor, respectively.

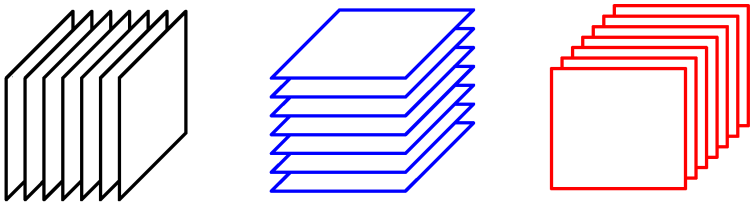


Figure 2.2: The vertical ($\underline{\boldsymbol{X}}(:,j,:)$), horizontal ($\underline{\boldsymbol{X}}(i,:,:)$), and frontal slabs ($\underline{\boldsymbol{X}}(:,:,k)$) of a third-order tensor, respectively.

parallel factor analysis (PARAFAC) [64]. The CPD elementwise representation can be written as:

$$\underline{\boldsymbol{X}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}(i,f)\boldsymbol{B}(j,f)\boldsymbol{C}(k,f), \tag{2.3}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_F] \in \mathbb{F}^{I\times F}$, $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_F] \in \mathbb{F}^{J\times F}$, $\boldsymbol{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_F] \in \mathbb{F}^{K\times F}$ are called the low rank factors of the tensor. A third-order tensor can be fully characterized by its latent factors, thus we adopt the notation

$$\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$$

to represent the tensor.

One nice property of tensors is that the CPD model is essentially unique even when F is much larger than $\max\{I, J, K\}$. This is a striking difference between tensors and matrices—the low-rank decomposition of a matrix is in general non unique. A generic result on the uniqueness of the CPD follows.

Theorem 2.1. *[40, p. 1019-1021] Let* $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ *with* $\boldsymbol{A} : I \times F$, $\boldsymbol{B} : J \times F$, *and* $\boldsymbol{C} : K \times F$. *Assume that* $\boldsymbol{A}$, $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *are drawn from some joint absolutely continuous*

distribution with respect to the Lebesgue measure in $\mathbb{F}^{(I+J+K)F}$*. Also assume* $I \geq J \geq K$ *without loss of generality. If* $F \leq 2^{\lfloor \log_2 J \rfloor + \lfloor \log_2 K \rfloor - 2}$*, then the decomposition of* $\underline{\boldsymbol{X}}$ *in terms of* $\boldsymbol{A}, \boldsymbol{B}$*, and* $\boldsymbol{C}$ *is essentially unique, almost surely. The notation* $\lfloor x \rfloor$ *is used for the largest integer that is less than or equal to* x*.*

In cases where the tensor rank F is less than or equal to one of the dimensions, the above conditions can be relaxed to the following:

Theorem 2.2. *[40] Let* $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ *with* $\boldsymbol{A} : I \times F$*,* $\boldsymbol{B} : J \times F$*, and* $\boldsymbol{C} : K \times F$*. Assume that* $\boldsymbol{A}$*,* $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *are drawn from some joint absolutely continuous distribution. Also assume* $I \geq J \geq K$ *without loss of generality and* $F \leq I$*. If* $F \leq \min(I, (J-1)(K-1))$*, then the decomposition of* $\underline{\boldsymbol{X}}$ *in terms of* $\boldsymbol{A}, \boldsymbol{B}$*, and* $\boldsymbol{C}$ *is essentially unique, almost surely.*

Here, essential uniqueness means that if $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}$ also satisfy $\underline{\boldsymbol{X}} = [\![\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{C}}]\!]$, then $\boldsymbol{A} = \tilde{\boldsymbol{A}}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1$, $\boldsymbol{B} = \tilde{\boldsymbol{B}}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2$, and $\boldsymbol{C} = \tilde{\boldsymbol{C}}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3$, where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}_i$ is a full rank diagonal matrix such that $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$. In Theorem 2.1 $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ are drawn from some joint absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{F}^{(I+J+K)F}$. For example, $[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]$ drawn from an i.i.d. Gaussian distribution over $\mathbb{F}^{(I+J+K)F}$ is absolutely continuous with respect to the respective Lebesgue measure, and so is any correlated Gaussian distribution with a non-singular covariance matrix. However, a Gaussian with a singular covariance does not fit this bill. As a more concrete example, consider two real zero-mean Gaussian random variables, X_1 and X_2. If $X_2 = cX_1$, and $X_1 \sim \mathcal{N}(0, 1)$, then their covariance matrix is $[1, c; c, c^2]$, which is singular, and the support of the joint distribution is a line, which has measure zero in $\mathbb{R}^2$.

As far as deterministic identifiability is concerned, we have:

Theorem 2.3. *[97] Let* $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ *with* $\boldsymbol{A} : I \times F$*,* $\boldsymbol{B} : J \times F$*, and* $\boldsymbol{C} : K \times F$*. The decomposition* $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ *is essentially unique with CP rank* F *if* $k_{\boldsymbol{A}} + k_{\boldsymbol{B}} + k_{\boldsymbol{C}} \geq 2F + 2$*.*

Here $k_{\boldsymbol{A}}$ denotes the Kruskal rank of a matrix, i.e., the largest integer $k_{\boldsymbol{A}}$ such that *any* $k_{\boldsymbol{A}}$ columns of $\boldsymbol{A}$ are linearly independent.

Note that all theorems presented in this thesis can be applied to real and complex tensors, i.e., $\underline{\boldsymbol{X}} \in \mathbb{R}^{I \times J \times K}$ or $\underline{\boldsymbol{X}} \in \mathbb{C}^{I \times J \times K}$, without any change. Whereas tensor rank generally depends on the field over which the decomposition is computed [143], our results apply to tensors in the real or complex field without any change. The reason is that Theorem 2.1 remains the same

for tensors with generic factors in real or complex field as stated in [40, p. 1021] and Kruskal's condition / proof, used in Theorem 2.3, is valid for both real and complex tensors.

2.2 Coupled Canonical Polyadic Decomposition

Another important tensor model is the coupled CPD. In coupled CPD we are interested in decomposing an array of tensors that share at least one common latent factor. In particular, consider a collection of N tensors:

$$\underline{\boldsymbol{X}}_n \in \mathbb{F}^{I \times J_n \times K_n},\ n \in \{1, \dots N\}. \tag{2.4}$$

The rank-F coupled CPD of $\{\underline{\boldsymbol{X}}_n\}$ can be expressed as:

$$\underline{\boldsymbol{X}}_n = [\![\boldsymbol{A}, \boldsymbol{B}_n, \boldsymbol{C}_n]\!],\ n \in \{1, \dots N\}, \tag{2.5}$$

where $\boldsymbol{A} \in \mathbb{F}^{I \times F}$ is the common factor and $\boldsymbol{B}_n \in \mathbb{F}^{J_n \times F},\ \boldsymbol{C}_n \in \mathbb{F}^{K_n \times F}$ are unshared factors. The coupled CPD is also unique under certain conditions, even if individual CPDs of $\boldsymbol{X}_n$ are not unique. Before we present the uniqueness Theorem for the coupled CPD model we first need to define the 2-nd compound matrix and a special matrix $\boldsymbol{G}$.

Definition 2.1. *[47, p. 861-862] The 2-nd compound matrix of an $J \times F$ matrix $\boldsymbol{B}$ is the $\frac{J(J-1)}{2} \times \frac{F(F-1)}{2}$ matrix containing the determinants of all 2×2 submatrices of $\boldsymbol{B}$ and is denoted by $C_2(\boldsymbol{B})$.*

Matrix $\boldsymbol{G}$ is defined as:

$$\boldsymbol{G} = \begin{bmatrix} C_2(\boldsymbol{C}_1) \odot C_2(\boldsymbol{B}_1) \\ \vdots \\ C_2(\boldsymbol{C}_N) \odot C_2(\boldsymbol{B}_N) \end{bmatrix} \ :\ \frac{1}{4}\sum_{n=1}^{N} J_n(J_n-1)K_n(K_n-1) \times \frac{1}{2}F(F-1).$$

In this thesis we will use the following uniqueness theorem for coupled CPD:

Theorem 2.4. *[153, p. 510] Consider the coupled CPD as expressed in* (2.5)*. The decomposition is essentially unique if:*

1. *$\boldsymbol{A}$ has full column rank,*

2. $\boldsymbol{G}$ *has full column rank.*

In the context of coupled CPD, essential uniqueness corresponds to $\boldsymbol{A}$ being unique and $\{\boldsymbol{B}_n,\ \boldsymbol{C}_n\}$ being identifiable up to column scaling and counter-scaling.

2.3 Tensor Algebra operations

Now we present two important operations of tensor algebra that are being used extensively throughout the thesis.

A tensor can be represented in a matrix form using the *matricization* operation. There are three common ways to matricize (or unfold) a third-order tensor, by stacking columns, rows, or fibers of the tensor to form a matrix. To be more precise let:

$$\underline{\boldsymbol{X}}(:,:,k) = \boldsymbol{X}^k \in \mathbb{F}^{I\times J}, \tag{2.6}$$

where $\boldsymbol{X}^k$ are the frontal slabs of tensor $\underline{\boldsymbol{X}}$. Then the mode-1, mode-2 and mode-3 unfoldings of $\underline{\boldsymbol{X}}$ can be cast as:

$$\boldsymbol{X}^{(\mathbf{1})} = \begin{bmatrix} \boldsymbol{X}^{1^T} \\ \boldsymbol{X}^{2^T} \\ \vdots \\ \boldsymbol{X}^{K^T} \end{bmatrix} \in \mathbb{F}^{JK\times I}, \tag{2.7}$$

$$\boldsymbol{X}^{(\mathbf{2})} = \begin{bmatrix} \boldsymbol{X}^{1} \\ \boldsymbol{X}^{2} \\ \vdots \\ \boldsymbol{X}^{K} \end{bmatrix} \in \mathbb{F}^{IK\times J}, \tag{2.8}$$

$$\boldsymbol{X}^{(\mathbf{3})} = \begin{bmatrix} \boldsymbol{X}^1(:,1), \boldsymbol{X}^2(:,1), \cdots, \boldsymbol{X}^K(:,1) \\ \boldsymbol{X}^1(:,2), \boldsymbol{X}^2(:,2), \cdots, \boldsymbol{X}^K(:,2) \\ \vdots \\ \boldsymbol{X}^1(:,J), \boldsymbol{X}^2(:,J), \cdots, \boldsymbol{X}^K(:,J) \end{bmatrix} \in \mathbb{F}^{IJ\times K}, \tag{2.9}$$

The superscript $(\cdot)$ denotes the mode according to which the unfolding is performed, e.g., is the superscript is (1) the matrisization is performed on the first mode of the tensor, i.e. columns are

stacked together. The CPD can also be reflected in the matricized version of tensor $\underline{\boldsymbol{X}}$. One can see that:

$$\boldsymbol{X}^{(\mathbf{1})} = (\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^T \tag{2.10}$$

$$\boldsymbol{X}^{(\mathbf{2})} = (\boldsymbol{C} \odot \boldsymbol{A})\boldsymbol{B}^T, \tag{2.11}$$

$$\boldsymbol{X}^{(\mathbf{3})} = (\boldsymbol{B} \odot \boldsymbol{A})\boldsymbol{C}^T, \tag{2.12}$$

where $\odot$ denotes the Khatri-Rao (column-wise Kronecker) product.

Another important operation in tensor analytics is the *mode product*. The mode product operator multiplies a matrix to a tensor in a single mode. A third order tensor has three modes (rows, columns, fibers), thus three different mode products are defined. A joint mode-1, mode-2, and mode-3 product of a third-order tensor is represented by the following notation:

$$\underline{\tilde{\boldsymbol{X}}} = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 \times_3 \boldsymbol{P}_3 \tag{2.13}$$

where "$\times_1$" denotes the operation that multiplies each column of $\underline{\boldsymbol{X}}$ with $\boldsymbol{P}_1$, "$\times_2$" denotes multiplying each row of $\underline{\boldsymbol{X}}$ with $\boldsymbol{P}_2$, and "$\times_3$" denotes multiplying each fiber of $\underline{\boldsymbol{X}}$ with $\boldsymbol{P}_3$. The mode product is reflected in the polyadic decomposition of the tensor, i.e., the outcome of (2.13) results in a tensor $\underline{\tilde{\boldsymbol{X}}}$ with polyadic decomposition:

$$\underline{\tilde{\boldsymbol{X}}} = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!],$$

The above decomposition is essentially unique under some conditions—this point will turn out to be crucial in the upcoming discussion.

Chapter 3

Hyperspectral Super-Resolution: A Coupled Tensor Factorization Approach

Hyperspectral super-resolution refers to the problem of fusing a hyperspectral image (HSI) and a multispectral image (MSI) to produce a super-resolution image (SRI) that admits fine spatial and spectral resolutions. State-of-the-art methods approach the problem via low-rank matrix approximations to the matricized HSI and MSI. These methods are effective to some extent, but a number of challenges remain. First, HSIs and MSIs are naturally third-order tensors (data "cubes") and thus matricization is prone to loss of structural information—which could degrade performance. Second, it is unclear whether these low-rank matrix-based fusion strategies can guarantee identifiability of the SRI under realistic assumptions. However, identifiability plays a pivotal role in estimation problems and usually has a significant impact on performance in practice. Third, the majority of the existing methods assume known (or easily estimated) degradation operators from the SRI to the corresponding HSI and MSI—which is hardly the case in practice. In this chapter, we propose to tackle the super-resolution problem from a tensor perspective. Specifically, we utilize the multidimensional structure of the HSI and MSI and propose two coupled tensor factorization frameworks that can effectively overcome the aforementioned issues. The proposed approaches guarantee the identifiability of the SRI under mild and realistic conditions. Furthermore, they work with little knowledge about the degradation

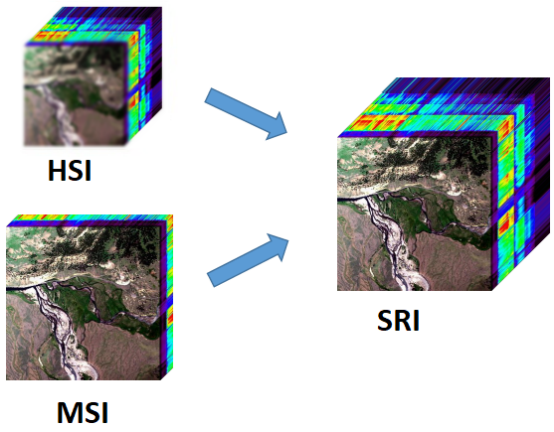


Figure 3.1: Illustration of the hyperspectral super-resolution task.

operators, which is clearly a favorable feature in practice. Simulations with real HSI's showcase the effectiveness of the proposed approaches. Part of this Chapter is published in [83–85].

3.1 Problem Statement and Background

Consider an HSI cube $\underline{\boldsymbol{Y}}_H \in \mathbb{R}^{I_H \times J_H \times K_H}$, where I_H and J_H denote the spatial dimensions and K_H denotes the number of spectral bands. Similarly, let $\underline{\boldsymbol{Y}}_M \in \mathbb{R}^{I_M \times J_M \times K_M}$ denote an MSI cube, where $I_M,\ J_M$ and K_M are the dimensions of the spatial and spectral domains, respectively. An HSI captures information over a broad range of the electromagnetic spectrum, usually involving hundreds of spectral bands/wavelengths. An MSI usually consists of pixels which are measured at less than 20 wavelengths; i.e., $K_M \ll K_H$ in general. On the other hand, MSIs have a much finer resolution in the spatial domain relative to HSIs—i.e., $I_H J_H \ll I_M J_M$ typically holds.

Hyperspectral super-resolution aims at integrating a pair of co-registered HSI and MSI, which describe the same target (e.g., a region on the ground), in order to form an SRI $\underline{\boldsymbol{Y}}_S \in \mathbb{R}^{I_M \times J_M \times K_H}$ that has the spatial resolution of the MSI and the spectral resolution of the HSI. The hyperspectral super-resolution task, illustrated in Fig. 3.1, is very well-motivated since both spectral and spatial information are rich and valuable to analytics and can benefit a number of applications such as image processing, remote sensing, geoscience, and food and medicine security, just to name a few.

3.1.1 Prior Art

HSR is a long-existing problem in remote sensing. For example, a lot of early works in the 1990s and 2000s studied the problem of hyperspectral pansharpening, which fuses an HSI and a panchromatic image to produce an SRI. This problem has a similar flavor as HSR; see a comprehensive review in [6, 109]. Existing pansharpening methods include component substitution (CS) [7, 36] and multiresolution analysis (MRA) [5, 107, 170], which stem from similar ideas that work by injecting details from the panchromatic image into the HSI. Attempts have been made to use pansharpening type methods for HSR. They are mainly based on wavelet techniques [60, 189] or try to generalize CS and MRA pansharpening algorithms for HSR purposes [138]. However, such methods were found to have difficulties with enhancing the spatial resolution of every hyperspectral band in practice [185].

Over the past few years, there has been a renewed interest for HSR, which is largely triggered by the advances in modern optimization and matrix factorization techniques. Numerous recent methods for HSR utilize low-rank matrix factorization models [101, 146, 163, 174, 176, 177, 179, 187]. The idea is to take advantage of the low-rank matrix structure of the matricized HSI and MSI. One such low-rank model is the so-called linear mixure model (LMM) which is widely employed for modeling hyperspectral/multispectral pixels. Under LMM, every spectral pixel of the HSI or MSI is modeled as a convex combination of the spectral signatures of several materials (or *endmembers*). This representation is physically intuitive and has enabled a large amount of hyperspectral unmixing algorithms [26, 38, 104, 112, 117]. More importantly, under LMM, all the pixels reside in a low-dimensional subspace spanned by a number of endmembers—which makes the matricized HSI/MSI of low rank. Several HSR approaches work under this model. For example, the works in [176, 179, 187] perform (coupled) low-rank factorization of the matricized HSI and MSI to estimate the spectral signatures of the endmembers (from HSI) and the corresponding high-resolution spatial distribution of the pixels (from MSI). Then the SRI is constructed by combining these two estimated matrices. A number of variants exist [146, 163, 174, 174], using different data representations and algorithms. Nevertheless, utilizing low-rank modeling and matricized HSI and MSI is the common feature of this line of work.

3.1.2 Matrix Factorization-based Approaches

The arguably most popular and effective existing HSR approaches are based on low-rank matrix factorization. Specifically, in [146, 163, 174, 176, 177, 179, 187], the matricized multiband images (i.e., SRI, HSI, MSI) are all modeled as low rank matrices, resulting from the linear mixure model (LMM) of the multiband pixels. To be specific, consider the matricized SRI as:

$$\boldsymbol{Y}_S = [\underline{\boldsymbol{Y}}_S(1,1,:), \ldots, \underline{\boldsymbol{Y}}_S(I_H, J_H, :)]^T \in \mathbb{R}^{I_M J_M \times K_H}, \tag{3.1}$$

where $\underline{\boldsymbol{Y}}_S(i,j,:) \in \mathbb{R}^{K_H}$ is a vector that is formed by taking the (i,j)th spectral pixel of the SRI. Under the LMM, a spectral pixel $\boldsymbol{Y}_S(:,\ell)$ is modeled as a weighted sum of the spectral signatures of several materials (or endmembers) that are present in the image:

$$\boldsymbol{Y}_S \approx \boldsymbol{S}_M \boldsymbol{E}_H^T, \tag{3.2}$$

where $\boldsymbol{E}_H \in \mathbb{R}^{K_H \times R}$ is the endmember matrix containing the spectral signatures of $R \ll \min\{I_M J_M, K_H\}$ materials in its R columns and $\boldsymbol{S}_M \in \mathbb{R}^{I_H J_H \times R}$ is the abundance matrix.

In order to tackle the HSR problem, existing work usually assumes that there exist two linear operators $\boldsymbol{P}_H \in \mathbb{R}^{I_H J_H \times I_M J_M}$ and $\boldsymbol{P}_M \in \mathbb{R}^{K_M \times K_H}$ such that $\boldsymbol{Y}_H = \boldsymbol{P}_H \boldsymbol{Y}_S$ and $\boldsymbol{Y}_M = \boldsymbol{Y}_S \boldsymbol{P}_M^T$. As a result, the matricized HSI is modeled as $\boldsymbol{Y}_H = (\boldsymbol{P}_H \boldsymbol{S}_M)\boldsymbol{E}_H^T$ and the matricized MSI as $\boldsymbol{Y}_M = \boldsymbol{S}_M(\boldsymbol{P}_M \boldsymbol{E}_H)^T$. Then, if $\boldsymbol{E}_H$ and $\boldsymbol{S}_M$ (or the range spaces of $\boldsymbol{E}$ and $\boldsymbol{S}$) can be estimated via jointly factoring $\boldsymbol{Y}_H$ and $\boldsymbol{Y}_M$ following the described model, the SRI is recovered following equation (3.2). This is the basic idea behind the low-rank factorization based HSR approaches.

3.1.3 Challenges.

The low rank matrix factorization approaches are effective to a certain extent and considered state of the art. However, three key theoretical and practical challenges remain.

First, as previously mentioned, multiband images are naturally data cubes that exhibit dependence across all the three dimensions. Using the matricized version of the 3D images is prone to loss of structural information. Some existing works tried to compensate this loss of information via promoting spatial smoothness (e.g., by adding total variation constraints on $\boldsymbol{S}_M$ or $\boldsymbol{S}_H$ [146]). This is a viable solution but several issues remain—e.g., this type of methods have to introduce a few more tuning parameters that are in general hard to determine.

In addition, merely using spatial smoothness still can not fully exploit the data structure that naturally represents the dependence across two spatial dimensions and one spectral dimension.

The second challenge is that recovering $\boldsymbol{Y}_S$ from the matricized HSI and MSI, i.e., $\boldsymbol{Y}_H$ and $\boldsymbol{Y}_M$, is an ill-posed inverse problem—an infinite number of solutions could exist. Making use of the low-rank modeling could help reduce the difficulty since it reduces the number of unknowns substantially—but there is still a lack of theoretical evidence that this approach could really recover $\boldsymbol{Y}_S$. One possible route for arguing identifiability is to connect the matrix factorization-based approaches to low-rank matrix sensing. However, this would require the degradation operators to be random [18, 30]. In our context, the degradation operators are highly structured, which means that known theory of matrix sensing cannot answer our question. The coupled factorization approaches with a variety of regularizations [146, 174, 176, 187] may help in practice—but currently lack theoretical guarantees. Note that identifiability is also important from a practical viewpoint, apart from theoretical. In particular, identifiability often serves as guidance for practitioners to select and design the appropriate solvers and algorithms—which have been proven very useful and powerful in pertinent problems, such as spectral unmixing [112]. Furthermore, it has been observed that, in a variety of problems (such as matrix and tensor decomposition), identifiability-guaranteed criteria usually entail much more stable numerical performance, e.g., being less sensitive to initialization compared to approaches that are lack of identifiability support [54, 143].

Another major concern is that the matrix-based methods commonly assume that the degradation operators $\boldsymbol{P}_H$ and $\boldsymbol{P}_M$ are accurately known or can be easily estimated, which is hardly the case in practice. The spectral response $\boldsymbol{P}_M$ can be relatively easy to model and estimate by comparing the spectral specifications of the hyperspectral and multispectral sensors. However, modeling the spatial operator can be rather difficult. One commonly used model assumes that the transformation from SRI to HSI is a combination of blurring by a Gaussian kernel and a downsampling process. This is of course a rough approximation and may be far from being accurate. Even if this assumption is approximately true, there is still a number of uncertainties such as the blurring function, the kernel size and the sampling offset. There are approaches in the literature, e.g., [146, 186], that attempt to estimate the degradation operators from data, but, again, these methods have to make a number of model assumptions regarding the degradation process, which can only approximately hold to some extent.

3.2 Degradation as Mode Product

In this section, we first reveal a nice connection between tensor mode products and the SRI-HSI/MSI degradation models. Building upon this connection, we will introduce coupled tensor factorization formulations and offer identifiability analyses next.

Let $\underline{\boldsymbol{Y}}_S \in \mathbb{R}^{I_M \times J_M \times K_H}$ be the target SRI we want to estimate. $\underline{\boldsymbol{Y}}_S$ admits a CPD with rank F, i.e.,

$$\begin{aligned} &\underline{\boldsymbol{Y}}_S = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!] \\ &\boldsymbol{A} : I_M \times F,\ \boldsymbol{B} : J_M \times F,\ \boldsymbol{C} : K_H \times F \end{aligned} \tag{3.3}$$

Also let $\underline{\boldsymbol{Y}}_H \in \mathbb{R}^{I_H \times J_H \times K_H}$ denote the corresponding HSI and $\underline{\boldsymbol{Y}}_M \in \mathbb{R}^{I_M \times J_M \times K_M}$ the MSI, respectively. Assume that there exist $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ such that the spatial degradation from the SRI to the HSI can be modeled as

$$\underline{\boldsymbol{Y}}_H(:,:,k) = \boldsymbol{P}_1 \underline{\boldsymbol{Y}}_S(:,:,k) \boldsymbol{P}_2^T,\ k = 1, \ldots, K_H, \tag{3.4}$$

where $\boldsymbol{P}_1 \in \mathbb{R}^{I_H \times I_M}$ and $\boldsymbol{P}_2 \in \mathbb{R}^{J_H \times J_M}$. The degradation model in Eq. (3.4) is intuitive: Blurring can be modeled as linear mixing of neighboring pixels under a certain kernel in both column and row dimensions. Downsampling can be viewed as linear compression—and the two procedures can be well modeled using a 'fat' matrix $\boldsymbol{P}_1$ and a 'tall' matrix $\boldsymbol{P}_2^T$ with appropriate kernels 'embedded' in the matrix elements. In fact, the model in (3.4) summarizes some popularly used blurring and downsampling models of the spatial degradation process. For example, in Appendix A.3, we show that the 2-D Gaussian blurring plus downsampling model that is widely adopted in the HSR literature [146, 163, 174, 176, 177, 179, 187] can be re-expressed in a form that is compatible with (3.4).

Under (3.4), it is straightforward to observe that the model described in (3.4) can be written as $\underline{\boldsymbol{Y}}_H = \underline{\boldsymbol{Y}}_S \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2$ (and thus $\boldsymbol{P}_H = \boldsymbol{P}_2 \otimes \boldsymbol{P}_1$ in the matricized form). Consequently, $\underline{\boldsymbol{Y}}_H$ can be represented in the following form:

$$\begin{aligned} &\underline{\boldsymbol{Y}}_H = \left[\!\!\left[\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \boldsymbol{C} \right]\!\!\right] \\ &\tilde{\boldsymbol{A}} = \boldsymbol{P}_1 \boldsymbol{A} : I_H \times F,\ \tilde{\boldsymbol{B}} = \boldsymbol{P}_2 \boldsymbol{B} : J_H \times F,\ \boldsymbol{C} : K_H \times F \end{aligned} \tag{3.5}$$

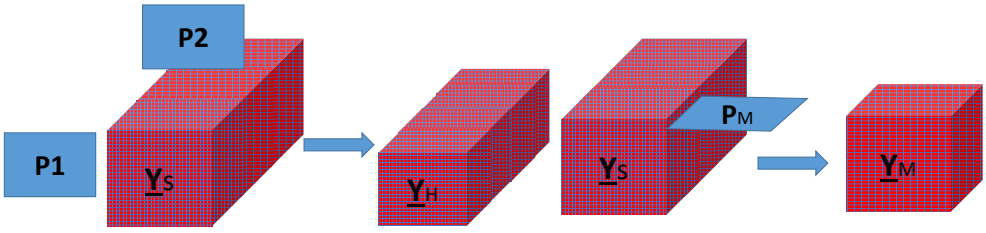


Figure 3.2: Illustration of degradation from the super-resolution image to the HSI and MSI, respectively.

In the matricized form, we have $\boldsymbol{Y}_H^{(3)} = \left(\tilde{\boldsymbol{B}} \odot \tilde{\boldsymbol{A}}\right) \boldsymbol{C}^T$.

The spectral degradation from the SRI to the MSI can be modeled as

$$\underline{\boldsymbol{Y}}_M(i,j,:) = \boldsymbol{P}_M \underline{\boldsymbol{Y}}_S(i,j,:) \ \ \forall i,j \tag{3.6}$$

where $\underline{\boldsymbol{Y}}_S(i,j,:) \in \mathbb{R}^K$ represents a fiber of the SRI and $\underline{\boldsymbol{Y}}_M(i,j,:) \in \mathbb{R}^K$ a fiber of the MSI, respectively. Matrix $\boldsymbol{P}_M \in \mathbb{R}^{K_M \times K_H}$ is usually modeled as a band-selection and averaging matrix. Eq. (3.6) is nothing but a mode-3 product operation, i.e., $\underline{\boldsymbol{Y}}_M = \underline{\boldsymbol{Y}}_S \times_3 \boldsymbol{P}_M$. Hence, $\underline{\boldsymbol{Y}}_M$ can be written as the following:

$$\begin{aligned} &\underline{\boldsymbol{Y}}_M = \left[\!\left[\boldsymbol{A}, \boldsymbol{B}, \tilde{\boldsymbol{C}}\right]\!\right] \\ &\boldsymbol{A} : I_M \times F, \ \boldsymbol{B} : J_M \times F, \ \tilde{\boldsymbol{C}} = \boldsymbol{P}_M \boldsymbol{C} : K_M \times F \end{aligned} \tag{3.7}$$

It is also readily seen that $\boldsymbol{Y}_M^{(3)} = (\boldsymbol{B} \odot \boldsymbol{A}) (\boldsymbol{P}_M \boldsymbol{C})^T$.

The discussed connection between HSR degradation and tensor mode products is visualized in Fig. 3.2. In retrospect, this connection is not very hard to reveal for someone versed in tensor algebra. However, the implication is very interesting and significant: If the "compressed" HSI and MSI tensors admit unique CPD models, then the SRI can be recovered. Intuitively, if one can identify the latent factors of $\underline{\boldsymbol{Y}}_M$ and $\underline{\boldsymbol{Y}}_H$ via CPD, respectively, then, the SRI can be reconstructed using $\underline{\boldsymbol{Y}}_S = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. This is of course a rough argument that must be fleshed out in a number of aspects, but it in fact reveals the major insight that leads to the first provable identifiability results for the HSR problem—as we will see in the next section. Another remark is that the connection between tensor mode product and spatial degradation holds based on the assumption that the horizontal and vertical blurring and downsampling applied to the SRI can be

represented as separable linear operators, which is reasonable when the overall blurring kernel is not skewed – and in practice blurring is usually isotropic, so our separability assumption holds.

3.3 Coupled Tensor Factorization for Super-resolution

Following the insights revealed in the previous section, we develop algorithms to handle the HSR problem in this section. We consider two cases: First, when the degradation operators are known, which follows the standard setups as the majority of matrix-based HSR works e.g. [101, 174, 176, 177, 187]. Second, when the spatial degradation operator is completely unknown, which is more realistic yet much more challenging. For both cases, we propose tensor-based algorithms and discuss the respective theoretical guarantees.

3.3.1 When $\boldsymbol{P}_H$ and $\boldsymbol{P}_M$ are known

Let us first consider the case where $\boldsymbol{P}_H$ and $\boldsymbol{P}_M$ are known. Recall that $\underline{\boldsymbol{Y}}_H = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]$ and $\underline{\boldsymbol{Y}}_M = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_M\boldsymbol{C}]\!]$, where $\tilde{\boldsymbol{A}} = \boldsymbol{P}_1\boldsymbol{A}$, $\tilde{\boldsymbol{B}} = \boldsymbol{P}_2\boldsymbol{B}$, and $\tilde{\boldsymbol{C}} = \boldsymbol{P}_M\boldsymbol{C}$. We wish to identify $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ from the HSI and MSI so that we can reconstruct the SRI. To this end, we propose to employ the following formulation:

$$\text{minimize}_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \ \|\underline{\boldsymbol{Y}}_H - [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2 + \lambda \|\underline{\boldsymbol{Y}}_M - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_M\boldsymbol{C}]\!]\|_F^2 . \tag{3.8}$$

In other words, we employ the above formulation to jointly decompose the HSI and MSI tensors to estimate $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$, where $\lambda > 0$ is a pre-selected parameter that weights the importance of each image in estimating $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$. After obtaining the estimates of $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$, the super-resolution tensor reconstruction is performed by

$$\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^{F} \hat{\boldsymbol{A}}(i,f)\hat{\boldsymbol{B}}(j,f)\hat{\boldsymbol{C}}(k,f).$$

The problem in (3.8) is a non-convex problem that is NP-hard in general. To tackle it, the *alternating optimization* (AO) framework is employed. Specifically one factor is updated at a time while keeping the rest fixed. Making use of the matricized forms of the HSI and MSI tensors, every step boils down to solving a Sylvester's equation—which is a classic convex quadratic problem and easy to handle (see details in Appendix B.2). The proposed *Super-resolution*

TEnsor-REcOnstruction (STEREO for short) is summarized in Algorithm 5.1.

Algorithm 3.1 STEREO

Initialization: $\lambda,\ F,\ \boldsymbol{A},\ \boldsymbol{B},\ \boldsymbol{C}$
repeat
$\boldsymbol{A} \leftarrow \arg\min_{\boldsymbol{A}} \|\boldsymbol{Y}_H^{(1)} - (\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})\boldsymbol{A}^T\boldsymbol{P}_1^T\|_F^2 + \lambda\|\boldsymbol{Y}_M^{(1)} - (\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^T\|_F^2$;
$\boldsymbol{B} \leftarrow \arg\min_{\boldsymbol{B}} \|\boldsymbol{Y}_H^{(2)} - (\boldsymbol{C} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{B}^T\boldsymbol{P}_2^T\|_F^2 + \lambda\|\boldsymbol{Y}_H^{(2)} - (\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{A})\boldsymbol{B}^T\|_F^2$;
$\boldsymbol{C} \leftarrow \arg\min_{\boldsymbol{C}} \|\boldsymbol{Y}_H^{(3)} - (\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{C}^T\|_F^2 + \lambda\|\boldsymbol{Y}_M^{(3)} - (\boldsymbol{B} \odot \boldsymbol{A})\boldsymbol{C}^T\boldsymbol{P}_M^T\|_F^2$;
until Some stopping criterion is met
Reconstruct $\underline{\boldsymbol{Y}}_S$ using $\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^F \boldsymbol{A}(i,f)\boldsymbol{B}(j,f)\boldsymbol{C}(k,f)$.

3.3.2 When $\boldsymbol{P}_H$ is unknown

We also consider the case where the spatial degradation operator $\boldsymbol{P}_H = \boldsymbol{P}_2 \otimes \boldsymbol{P}_1$ is completely unknown. As previously explained, considering this scenario is very well-motivated: Although $\boldsymbol{P}_M$ is relatively easy to model since it is well recognized as a uniform spectral response function[1] [19], the spatial degradation operator is quite hard to accurately model and estimate. Even when the operation is known as a combination of blurring and downsampling, the hyperparameters such as the blurring kernel type, the kernel size and the downsampling offset are hardly known in practice. To circumvent this, we propose to employ the following estimator for $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$:

$$\text{minimize}_{\boldsymbol{A},\boldsymbol{B},\tilde{\boldsymbol{A}},\tilde{\boldsymbol{B}},\boldsymbol{C}} \ \left\|\underline{\boldsymbol{Y}}_H - \left[\!\!\left[\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \boldsymbol{C}\right]\!\!\right]\right\|_F^2 + \lambda \left\|\underline{\boldsymbol{Y}}_M - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_M\boldsymbol{C}]\!]\right\|_F^2. \tag{3.9}$$

Problem (3.9) is harder than Problem (3.8) since it has more unknowns to estimate (as will be reflected in the theoretical analysis in the next subsection). Nevertheless, this problem can still be tackled using AO as we applied for handling Problem (3.8). The Blind STEREO algorithm that handles problem (3.9) is described in Algorithm 3.2. We use 'Blind' in the algorithm's name to distinguish it with STEREO, since Algorithm 3.2 is *spatially blind*—i.e., it does not need any prior knowledge on the spatial degradation operator.

Remark 3.1. Algorithms 1 and 2 are both instances of the *block coordinate descent* (BCD) optimization strategy. The algorithms decrease the objective in every iteration, and thus the produced cost value sequence converges. One subtle point here is that the solution sequence

[1] A reasonable estimate of $\boldsymbol{P}_M$ can usually be obtained after comparing the hyperspectral and multispectral specifications, i.e the employed wavelengths of the HSI and MSI cameras.

Algorithm 3.2 `Blind STEREO`

Initialization: λ, F, $\boldsymbol{A}$, $\boldsymbol{B}$, $\tilde{\boldsymbol{A}}$, $\tilde{\boldsymbol{B}}$
repeat
$\boldsymbol{C} \leftarrow \arg\min_{\boldsymbol{C}} \|\boldsymbol{Y}_H^{(3)} - (\tilde{\boldsymbol{B}} \odot \tilde{\boldsymbol{A}})\boldsymbol{C}^T\|_F^2 + \lambda\|\boldsymbol{Y}_M^{(3)} - (\boldsymbol{B} \odot \boldsymbol{A})\boldsymbol{C}^T\mathbf{P}_3^T\|_F^2$;
$\tilde{\boldsymbol{A}} \leftarrow \arg\min_{\tilde{\boldsymbol{A}}} \|\boldsymbol{Y}_H^{(1)} - (\boldsymbol{C} \odot \tilde{\boldsymbol{B}})\tilde{\boldsymbol{A}}^T\|_F^2$;
$\tilde{\boldsymbol{B}} \leftarrow \arg\min_{\tilde{\boldsymbol{B}}} \|\boldsymbol{Y}_H^{(2)} - (\boldsymbol{C} \odot \tilde{\boldsymbol{A}})\tilde{\boldsymbol{B}}^T\|_F^2$;
$\boldsymbol{A} \leftarrow \arg\min_{\boldsymbol{A}} \|\boldsymbol{Y}_M^{(1)} - (\boldsymbol{P}_3\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^T\|_F^2$;
$\boldsymbol{B} \leftarrow \arg\min_{\boldsymbol{B}} \|\boldsymbol{Y}_M^{(2)} - (\boldsymbol{P}_3\boldsymbol{C} \odot \boldsymbol{A})\boldsymbol{B}^T\|_F^2$;
until Some stopping criterion is met
Reconstruct $\underline{\boldsymbol{Y}}_S$ using $\underline{\hat{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^F \boldsymbol{A}(i,f)\boldsymbol{B}(j,f)\boldsymbol{C}(k,f)$.

may not converge, since the subproblems may have multiple solutions [25]. According to our extensive simulations, this barely affects performance. Nevertheless, if one wishes to fix this theoretical issue, one simple method as suggested in [133] is adding a proximal term such as $\rho^k\|\boldsymbol{A} - \boldsymbol{A}^{k-1}\|_F^2$ in the kth iteration to the cost function of the subproblems—which does not increase the difficulty of the subproblems but makes the cost function strongly convex. Consequently, one can show that every limit point of the solution sequence is a stationary point following the block successive upperbound minimization (BSUM) framework [133].

3.3.3 Identifiability Analysis

In this section, we present the identifiability analysis of the proposed approaches. Unlike the matrix factorization-based approaches that mostly have no identifiability characterization of the methods, we show that the proposed estimators can guarantee identifiability of the super-resolution tensor under realistic conditions.

To proceed, let us first consider the following important lemma:

Lemma 3.1. *Let $\tilde{\boldsymbol{Z}} = \boldsymbol{Q}\boldsymbol{Z}$, where the elements of $\boldsymbol{Z}$ are drawn from an absolutely continuous joint distribution with respect to the Lebesgue measure in $\mathbb{R}^{IF}$ and $\boldsymbol{Q} \in \mathbb{R}^{I' \times I}$ is deterministic with full row rank. Then the joint distribution of the elements in $\tilde{\boldsymbol{Z}}$ is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}^{I'F}$*

Proof. Define $\tilde{\boldsymbol{z}} := \text{vec}(\tilde{\boldsymbol{Z}})$ and $\boldsymbol{z} := \text{vec}(\boldsymbol{Z})$. Then, we have

$$\tilde{\boldsymbol{z}} = \text{vec}(\boldsymbol{Q}\boldsymbol{Z}) = \text{vec}(\boldsymbol{Q}\boldsymbol{Z}\boldsymbol{I}) = (\boldsymbol{I} \otimes \boldsymbol{Q})\boldsymbol{z}.$$

Now, define $\boldsymbol{P} = \boldsymbol{I} \otimes \boldsymbol{Q} \in \mathbb{R}^{I'F \times IF}$, which is a 'fat' matrix since $I'F \leq IF$. By properties of the Kronecker product, we have

$$\text{rank}(\boldsymbol{P}) = \text{rank}(\boldsymbol{Q})\text{rank}(\boldsymbol{I}) = I'F.$$

Furthermore, let $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ denote the full-size singular value decomposition (SVD) of $\boldsymbol{P}$, where $\boldsymbol{U} \in \mathbb{R}^{I'F \times I'F}$, $\boldsymbol{V} \in \mathbb{R}^{IF \times IF}$ are orthonormal matrices and $\boldsymbol{\Sigma} \in \mathbb{R}^{I'F \times IF}$ consists of a diagonal submatrix as its first $I'F$ columns (which holds the singular values as the diagonal elements) and an all-zero submatrix, i.e.,

$$\boldsymbol{\Sigma} = [\text{diag}(\sigma_1, \ldots, \sigma_{I'F}), \boldsymbol{0}] \in \mathbb{R}^{I'F \times IF}.$$

Consider $\boldsymbol{z}_V = \boldsymbol{V}^T\boldsymbol{z}$ and let $f_Z(\boldsymbol{z})$ denote the joint probability density function (PDF) of $\boldsymbol{z}$ with respect to the Lebesgue measure in $\mathbb{R}^{IF}$. The random vector $\boldsymbol{z}_V$ is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}^{IF}$, since the Lebesgue measure is invariant under unitary transformations [78] and the PDF of $\boldsymbol{z}_V$ takes the following form [184]:

$$f_{Z_V}(\boldsymbol{z}_V) = f_Z(\boldsymbol{V}\boldsymbol{z}).$$

Now, consider $\boldsymbol{z}_\Sigma = \boldsymbol{\Sigma}\boldsymbol{z}_V$. This matrix-vector product selects and positively weights the first $I'F$ random variables in $\boldsymbol{z}_V$, i.e.,

$$\boldsymbol{z}_\Sigma = \text{diag}(\boldsymbol{\sigma})\tilde{\boldsymbol{z}}_V,\ \tilde{\boldsymbol{z}}_V = \boldsymbol{z}_V(1 : I'F) \in \mathbb{R}^{I'F},$$

where $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_{I'F}]^T$. The above product does not hurt the continuity of the joint distribution of the random variables in $\tilde{\boldsymbol{z}}_V$ (since the joint PDF of $\boldsymbol{z}_V$ can be obtained via marginalizing the joint PDF of $\boldsymbol{z}$), and thus $\boldsymbol{z}_\Sigma$ is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}^{I_H F}$. Finally, consider $\tilde{\boldsymbol{z}} = \boldsymbol{U}\boldsymbol{z}_\Sigma$. Again, $\tilde{\boldsymbol{z}}$ is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}^{IF}$, since $\boldsymbol{U}$ is a unitary transformation and the PDF is

$$f_{\tilde{Z}}(\tilde{\boldsymbol{z}}) = f_{Z_\Sigma}(\boldsymbol{U}^T\boldsymbol{z}_\Sigma),$$

. □

With Lemma 3.1 in our hands, we can show identifiability of the formulations in (3.8)-(3.9). To see this, let us first consider the case where the spatial and spectral degradation operators are known. Regarding the identifiability of the SRI cube, let us make some model assumptions to simplify the analysis. We first assume that $I_M \geq J_M \geq K_M$ since K_M is usually quite small (i.e., usually being a single digit) and $I_H \geq J_H$. The number of hyperspectral bands, i.e., K_H, could be larger than I_H and J_H, depending on how large is the spatial area that we are interested in. Bearing these in mind, we have the following theorem:

Theorem 3.1. *Assume that* $\underline{\boldsymbol{Y}}_S = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$, $\underline{\boldsymbol{Y}}_H = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]$ *and* $\underline{\boldsymbol{Y}}_M = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_M\boldsymbol{C}]\!]$. *In addition, assume that* $I_M \geq J_M \geq K_M$, *that* $\boldsymbol{A}$, $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *are drawn from some absolutely continuous distribution with respect to the Lebesgue measure in* $\mathbb{R}^{(I_M+J_M+K_H)F}$, *that* $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ *and* $\boldsymbol{P}_M$ *have full rank, and that* $(\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star)$ *is an optimal solution to Problem* (3.8) *(whose corresponding value of the cost function is 0) when* $\lambda > 0$. *Then,*

$$\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$$

recovers the ground-truth $\underline{\boldsymbol{Y}}_S$ *almost surely if*

$$F \leq \min\{2^{\lfloor \log_2(K_M J_M)\rfloor - 2}, I_H J_H\}.$$

The proof is relegated to Appendix A.1. We should mention that the above bound is proven by judiciously combining Theorem 2.1, Lemma 3.1, and the problem structure—and the bound can be improved if $I_M \geq F$ holds. Specifically, we have:

Corollary 3.1. *Under the same assumptions as in Theorem 3.1, if* $I_M \geq F$, *we have that* $\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ *recovers the ground-truth* $\underline{\boldsymbol{Y}}_S$ *almost surely if* $F \leq \min\{(J_M-1)(K_M-1), I_H J_H\}$.

The proof of Corollary 3.1 is almost the same as that of Theorem 3.1. The only difference is that Theorem 2.2 (instead of Theorem 2.1) is invoked. The proof is omitted due to space limitation.

For the case where $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are unknown, we have the following theorem:

Theorem 3.2. *Assume the same generative model as in Theorem 3.1, that* $I_M \geq J_M \geq K_M$ *and* $I_H \geq J_H$, *that* $I_M J_M \geq I_H J_H$ *and* $K_M \leq K_H$, *and that* $(\tilde{\boldsymbol{A}}^\star, \tilde{\boldsymbol{B}}^\star, \boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star)$ *is an*

optimal solution to Problem (3.9) *(whose corresponding value of the cost function is 0), when* $\lambda > 0$*. Then,* $\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ *recovers the ground-truth* $\underline{\boldsymbol{Y}}_S$ *almost surely,*

1. *if* $F \leq \min\{2^{\lfloor \gamma_1 \rfloor - 2}, 2^{\lfloor \gamma_2 \rfloor - 2}\}$*, where* $\gamma_1 = \log_2(J_M K_M)$ *and* $\gamma_2 = \log_2(J_H K_H)$*, when* $I_H \geq K_H$*; and*

2. *if* $F \leq \min\{2^{\lfloor \gamma_1 \rfloor - 2}, 2^{\lfloor \gamma_2 \rfloor - 2}\}$*, where* $\gamma_1 = \log_2(J_M K_M)$ *and* $\gamma_2 = \log_2(I_H J_H)$*, when* $J_H < K_H$*.*

Note that if $I_M \geq F$, $2^{\lfloor \gamma_1 \rfloor - 2}$ can be replaced by $(J_M - 1)(K_M - 1)$. Similarly, if $I_H \geq F$, $2^{\lfloor \gamma_2 \rfloor - 2}$ can be replaced by $(J_H - 1)(\min\{I_H, K_H\} - 1)$. The proof of Theorem 3.2 is relegated to Appendix A.2. Note that Theorem 3.1 only requires that the CPD of the MSI tensor is unique, and has more relaxed conditions compared to those in Theorem 3.2, which needs the CPDs of both the HSI and MSI to be unique. This echoes our comment that Problem (3.9) is harder than Problem (3.8), since the former works under the case where one knows less about the model.

To have some concrete sense about the theorems, consider the case where we intend to reconstruct an SRI of size $600 \times 520 \times 180$ from an HSI of size $150 \times 130 \times 180$ and an MSI of size $600 \times 520 \times 8$. By Theorems 3.1 - 3.2, the identifiability of the SRI is guaranteed if the CPD rank of the SRI tensor satisfies $F \leq 1024$. This is in general easy to satisfy (approximately) in practice. To verify this, in Tables 3.1 - 3.4, we use a CPD model to reconstruct real-world hyperspectral images captured by the AVIRIS [162] and the ROSIS [100] hyperspectral sensors. One can see that the fitting error, defined as $\|\hat{\underline{\boldsymbol{Y}}} - \underline{\boldsymbol{Y}}\|_F / \|\underline{\boldsymbol{Y}}\|_F$ (where $\hat{\underline{\boldsymbol{Y}}}$ and $\underline{\boldsymbol{Y}}$ are the CPD model approximated HSI and the original HSI, respectively), is rather small (in the order of 10^{-2}) for all tested ranks (under all these ranks the CPD is unique). Tables 3.1 - 3.4 show that using an identifiable CPD model to approximate real-world hyperspectral/multispectral images is very reasonable.

Table 3.1: The NMSE of using a CPD model to approximate a subimage of the AVIRIS Cuprite data that is of size $512 \times 614 \times 187$.

rank	300	400	500	600	700	800
fitting error	0.0166	0.014	0.0125	0.0115	0.0108	0.0102

Remark 3.2. Both of our algorithms can be understood as *coupled tensor factorization* (CTF). CTF was studied in the literature in various forms, e.g., [51, 153]. Nevertheless, [51] is not

Table 3.2: The NMSE of using a CPD model to approximate a subimage of the Pavia University data that is of size $608 \times 336 \times 103$.

rank	300	400	500	600	700	800
fitting error	0.0635	0.0491	0.0403	0.0349	0.0311	0.0283

Table 3.3: The NMSE of using a CPD model to approximate a subimage of the Salinas data that is of size $80 \times 84 \times 204$.

rank	20	50	100	200	300
fitting error	0.0385	0.0145	0.0065	0.0047	0.0038

Table 3.4: The NMSE of using a CPD model to approximate a subimage of the Indian Pines data that is of size $144 \times 144 \times 200$.

rank	50	100	200	300	400	500
fitting error	0.0435	0.0334	0.0276	0.0247	0.0225	0.0205

concerned with identifiability issues but a computational framework under specific noise types. Reference [153] considers identifiability of coupled tensor decomposition with no linear operators (e.g., $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$) involved, which hence does not cover the results in Theorems 3.1-3.2.

3.4 Combining low-rank Tensor and Matrix structure

In the previous section, we proposed a coupled tensor factorization approach. In this section we propose a hybrid approach that combines the benefits of both tensor and matrix models. Specifically we model the super-resolution image as a low-rank tensor, while simultaneously imposing low rank matrix structure as the LMM suggests. The proposed hybrid model is identifiable and enjoys the nice properties of both models. Furthermore, we introduce a brand new hybrid algorithm, which is very appealing due to its simplicity, accuracy, and ability to work without any knowledge of the spatial degradation operator.

3.4.1 The Hybrid model

Recall that, the SRI is a tensor that admits a CPD $\underline{\boldsymbol{Y}}_S = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ of rank F. Moreover, following the LMM, the mode 3 unfolding of the SRI, $\boldsymbol{Y}_S^{(3)}$, exhibits low rank matrix structure of rank R. This is reflected in the singular value decomposition (SVD) of $\boldsymbol{Y}_S^{(3)} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The

columns of $\boldsymbol{V} \in \mathbb{R}^{K_H \times R}$ are the right singular vectors of $\boldsymbol{Y}_S^{(3)}$ and give an orthogonal basis for the fiberspace of tensor $\underline{\boldsymbol{Y}}_S$. Then one can without loss of generality compress the original super-resolution tensor $\underline{\boldsymbol{Y}_S^{(3)}} \in \mathbb{R}^{I_M \times J_M \times K_H}$ to a super-resolution core tensor $\underline{\boldsymbol{Z}}_S \in \mathbb{R}^{I_M \times J_M \times R}$, as $\underline{\boldsymbol{Z}}_S = \underline{\boldsymbol{Y}}_S \times_3 \boldsymbol{V}^T$. The CPD model of the core tensor is:

$$\underline{\boldsymbol{Z}}_S = [\![\boldsymbol{A}, \boldsymbol{B}, \bar{\boldsymbol{C}}]\!] \tag{3.10}$$

where $\bar{\boldsymbol{C}} = \boldsymbol{V}^T\boldsymbol{C} \in \mathbb{R}^{R \times F}$. Note that one can always recover $\underline{\boldsymbol{Y}}_S$ as $\underline{\boldsymbol{Y}}_S = \underline{\boldsymbol{Z}}_S \times_3 \boldsymbol{V}$ and $\boldsymbol{C}$ from $\boldsymbol{C} = \boldsymbol{V}\bar{\boldsymbol{C}}$, since $\underline{\boldsymbol{Y}}_S,\ \boldsymbol{C}$ live in a low dimensional subspace defined by $\boldsymbol{V}$; see [48, 143] for details. The HSI is related to SRI via (3.4). Therefore $\underline{\boldsymbol{Y}}_S,\ \underline{\boldsymbol{Y}}_H$ share the same fiberspace $\boldsymbol{V}$, which can be computed by the SVD of $\boldsymbol{Y}_H^{(3)}$. As a result, one may, without loss of generality, compress the original hyperspectral tensor $\underline{\boldsymbol{Y}}_H$ to a hyperspectral core tensor $\underline{\boldsymbol{Z}}_H \in \mathbb{R}^{I_H \times J_H \times R}$ as $\underline{\boldsymbol{Z}}_H = \underline{\boldsymbol{Y}}_H \times_3 \boldsymbol{V}^T = \underline{\boldsymbol{Y}}_S \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 \times_3 \boldsymbol{V}^T$. As we will see, this compression enables a very efficient recovery algorithm. The CPD model of $\underline{\boldsymbol{Z}}_H$ is then:

$$\underline{\boldsymbol{Z}}_H = \left[\!\!\left[\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \bar{\boldsymbol{C}}\right]\!\!\right], \tag{3.11}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{P}_1\boldsymbol{A} \in \mathbb{R}^{I_H \times F}$, $\tilde{\boldsymbol{B}} = \boldsymbol{P}_2\boldsymbol{B} \in \mathbb{R}^{J_H \times F}$.

Regarding the relation between the MSI and the core SRI, $\underline{\boldsymbol{Y}}_M = \underline{\boldsymbol{Y}}_S \times_3 \boldsymbol{P}_M$ and $\underline{\boldsymbol{Y}}_S = \underline{\boldsymbol{Z}}_S \times_3 \boldsymbol{V}$. Thus, $\underline{\boldsymbol{Y}}_M = \underline{\boldsymbol{Z}}_S \times_3 \bar{\boldsymbol{P}}_M$, where $\bar{\boldsymbol{P}}_M = \boldsymbol{P}_M\boldsymbol{V} \in \mathbb{R}^{K_M \times R}$. Consequently the CPD model of the MSI can be casted as:

$$\underline{\boldsymbol{Y}}_M = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_M\boldsymbol{C}]\!] = [\![\boldsymbol{A}, \boldsymbol{B}, \bar{\boldsymbol{P}}_M\bar{\boldsymbol{C}}]\!], \tag{3.12}$$

Overall, the hybrid model describes the SRI and HSI by a CPD model that admits a low rank matrix structure in the third mode (fiberspace). As far as the MSI is concerned, it is also described by a CPD model and the low rank matrix structure of the third mode is reflected in the spatial degradation operator which is transformed to $\bar{\boldsymbol{P}}_M = \boldsymbol{P}_M\boldsymbol{V}$. The (hybrid model based) super-resolution task is performed by identifying $\boldsymbol{A}$, $\boldsymbol{B}$, $\bar{\boldsymbol{C}}$ and $\boldsymbol{C} = \boldsymbol{V}\bar{\boldsymbol{C}}$.

3.4.2 Super-resolution Cube Algorithm (`SCUBA`)

Taking a closer look at the hybrid model we observe that we are able to compress the spectral dimension of the HSI from K_H to R without loss of generality. This compression also affects

the MSI model by transforming the spectral response from $\boldsymbol{P}_M \in \mathbb{R}^{K_M \times K_H}$ to $\bar{\boldsymbol{P}}_M = \boldsymbol{P}_M \boldsymbol{V} \in \mathbb{R}^{K_M \times R}$. In practice, the number of multispectral bands is usually between $K_M = 4$ and $K_M = 8$. The number of endmembers, R, on the other hand, depends on the size and type of the image, but is usually less than 20. While this case can be successfully handled using coupled tensor factorization, here we would like to point out a different and quite intriguing possibility. Namely, for $R \leq K_M$, SRI reconstruction can be accomplished in a simple and appealing way and under relaxed identifiability conditions – even if the spatial degradation operator is non-separable and completely unknown.

Let $\underline{\boldsymbol{Y}}_M$ denote the MSI with CPD $\underline{\boldsymbol{Y}}_M = \left[\!\left[\boldsymbol{A}, \boldsymbol{B}, \tilde{\boldsymbol{C}} \right]\!\right]$, where $\tilde{\boldsymbol{C}} = \bar{\boldsymbol{P}}_M \bar{\boldsymbol{C}}$. Also let $\boldsymbol{V} \in \mathbb{R}^{K_H \times R}$ be the basis of the hyperspectral fiberspace computed via SVD of $\boldsymbol{Y}_H^{(3)}$. If $R \leq K_M$, $\bar{\boldsymbol{C}}$ can be computed by solving the overdetermined system $\tilde{\boldsymbol{C}} = \bar{\boldsymbol{P}}_M \bar{\boldsymbol{C}}$, and consequently $\boldsymbol{C}$ can be obtained as $\boldsymbol{C} = \boldsymbol{V}\bar{\boldsymbol{C}}$. Note that the connection between the HSI LMM and the MSI tensor model renders the relation between $\boldsymbol{C}$ and $\tilde{\boldsymbol{C}}$ from highly under-determined to over-determined and is the key to the proposed algorithm. The procedure is summarized in the following steps:

$$\left[\!\left[\boldsymbol{A}, \boldsymbol{B}, \tilde{\boldsymbol{C}} \right]\!\right] \leftarrow \texttt{CPD}(\underline{\boldsymbol{Y}}_M) \tag{3.13a}$$

$$\boldsymbol{V} \leftarrow \texttt{SVD}(\boldsymbol{Y}_H^{(3)}) \tag{3.13b}$$

$$\boldsymbol{C} = \boldsymbol{V}\bar{\boldsymbol{P}}_M^{\dagger}\tilde{\boldsymbol{C}} \tag{3.13c}$$

$$\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}(i,f)\boldsymbol{B}(j,f)\boldsymbol{C}(k,f), \tag{3.13d}$$

where $\dagger$ denotes Moore-Penrose pseudoinverse. The caveat is that $R \leq K_M$ is restrictive in practice. The engineering solution is to judiciously choose, e.g., $8 \times 8 \times K$ blocks of the original image tensor, similar to what is done in JPEG image compression. Small spatial patches typically contain few endmembers, hence $R \leq K_M$ holds over each patch. Also note that smaller-size tensors typically exhibit smaller tensor rank, allowing us to use a smaller F per sub-tensor. The proposed *super-resolution cube algorithm* (`SCUBA` for short) is summarized in Algorithm 3.3. In the algorithm $\underline{\boldsymbol{Y}}_{S_l}$, $\underline{\boldsymbol{Y}}_{M_l}$, $\underline{\boldsymbol{Y}}_{H_l}$ denote the l-th MSI, HSI cube respectively.

3.4.3 `SCUBA` Identifiability

While blocking may introduce artifacts in highly compressed JPEG images, this is not a concern in our context when the decomposition of each sub-tensor is identifiable – since we can then

Algorithm 3.3 SCUBA

Judiciously **cut** $\underline{\boldsymbol{Y}}_M$, $\underline{\boldsymbol{Y}}_H$ into L cubes.
for $l = 1$ **to** L **do**

$$\left[\!\left[\boldsymbol{A}, \boldsymbol{B}, \tilde{\boldsymbol{C}}\right]\!\right] \leftarrow \texttt{CPD}(\underline{\boldsymbol{Y}}_{M_l})$$
$$\boldsymbol{V} \leftarrow \texttt{SVD}(\boldsymbol{Y}_{H_l}^{(3)})$$
$$\boldsymbol{C} = \boldsymbol{V}\bar{\mathbf{P}}_M^{\dagger}\tilde{\boldsymbol{C}}$$
$$\hat{\underline{\boldsymbol{Y}}}_{S_l} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$$

end for

provably reconstruct each super-resolution sub-tensor independently of its neighbors.

Theorem 3.3. *Let* $\underline{\boldsymbol{Y}}_M = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_M\boldsymbol{C}]\!]$ *and* $\boldsymbol{Y}_H^{(3)} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}}$*, where* $R \leq K_M$*. Assume without loss of generality that* $I_M \geq J_M \geq K_M$*. Also assume that* $\boldsymbol{A}$*,* $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *are jointly drawn from an absolutely continuous distribution, and that* $\boldsymbol{P}_H$ *and* $\boldsymbol{P}_M$ *have full rank. Let* $(\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star)$ *denote a solution to* (3.13a)-(3.13c)*. Then,* $\hat{\underline{\boldsymbol{Y}}}_S(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ *recovers the ground-truth* $\underline{\boldsymbol{Y}}_S$ *almost surely if* $F \leq 2^{\lfloor \log_2 J_M \rfloor + \lfloor \log_2 K_M \rfloor - 2}$.

As a concrete example, consider the reconstruction of a SRI of size $128 \times 128 \times 178$ from an MSI of size $128 \times 128 \times 8$ and an HSI of size $32 \times 32 \times 178$. Theorem 3.3 states that reconstruction is guaranteed if the rank of the MSI satisfies $F \leq 256$. The proof of Theorem 3.3 uses Theorem 2.1 to characterize the solution of (3.13a) and is similar to the proof of Therorem 3.1 which can be found in Appendix A.1.

3.5 Simulations

In this section, we showcase the effectiveness of the proposed HSR frameworks using numerical experiments. We generate simulated HSIs and MSIs following the Wald's protocol [172]. In Wald's protocol, the SRI-HSI degradation consists of spatial blurring by a convolutional kernel and a downsampling procedure. In order to obtain an MSI from an SRI, the spectral specifications of the multispectral sensor are used, which in our experiments are taken from the LANDSAT [1] or the QuickBird sensor [2]. The LANDSAT sensor produces a 6-band MSI by capturing information in the following spectral bands: Blue (450 - 520 nm), Green (520 - 600 nm), Red (630 - 690 nm), Near-IR (760 - 900 nm), Shortwave-IR1 (1550 - 1750 nm), Shortwave-IR2 (2080 - 2350 nm), whereas the QuickBird sensor produces a 4-band MSI in Blue (430 - 545 nm),

Green (466 - 620 nm), Red (590 - 710 nm) and Near-IR (715 - 918 nm). Then, the specifications of the available SRI, which span the spectrum from 400nm to 2500nm in our experiments, are compared with the multispectral sensor bands to form spectral response matrix $\boldsymbol{P}_M$ and thus the tested MSI images. To be more precise, $\boldsymbol{P}_M$ is a selection-averaging matrix which acts on the common wavelengths of the SRI and MSI.

Baselines

A set of baseline algorithms are employed for comparison, namely, `FUSE` [177], `FUSE-Sparse` [174, 175], `FUMI` [176], `HySure` [146] and `CNMF` [187]—which have all demonstrated competitive performance in the literature. All simulations are performed in MATLAB on a Linux server with 3.6GHz cores and 32GB RAM. We propose two CPD based algorithms, namely, `TenRec` and `Blind TenRec`, to cleverly initialize `STEREO`, `Blind STEREO` and each sub-tensor update in `SCUBA`. The idea is to compute the CPD of $\underline{\boldsymbol{Y}}_M$ in order to retrieve $\boldsymbol{A},\ \boldsymbol{B}$ and then solve a least squares problem to obtain $\boldsymbol{C}$. This way, an initial guess of the latent factors can be obtained. Consequently, the operational time of the algorithms can be substantially reduced, and an enhanced super-resolution accuracy is empirically observed. Detailed description of the initialization techniques are relegated to Appendix B.1. The CPD part performed in `TenRec` and `Blind TenRec` is computed using `Tensorlab` [166] with 25 iterations at maximum. In all the simulations, we fix $\lambda = 1$ and run `STEREO` for 10 iterations.

Evaluation

We largely follow the established conventions in the HSR literature for evaluating the results. Specifically, we adopt several intuitive metrics introduced in [6]. The first metric is *cross correlation (CC)* that is defined as

$$\texttt{CC} = \sum_{k=1}^{K} \rho(\underline{\boldsymbol{Y}_S}(:,:,k), \underline{\hat{\boldsymbol{Y}}_S}(:,:,k))$$

where ρ is the pearson correlation coefficient between the estimated and the reference slabs (i.e., $\underline{\hat{\boldsymbol{Y}}_S}(:,:,k)$ and $\underline{\boldsymbol{Y}_S}(:,:,k)$, respectively). `CC` is a score between 0 and 1, and 1 corresponds to the best estimation result. The second metric is called *spectral angle mapper (SAM)*, whose

definition is as follows:

$$\texttt{SAM} = \sum_{n=1}^{IJ} \arccos\left(\frac{\boldsymbol{Y}_S^{(3)}(n,:)\hat{\boldsymbol{Y}}_S^{(3)}(n,:)^T}{\|\boldsymbol{Y}_S^{(3)}(n,:)\|_2\|\hat{\boldsymbol{Y}}_S^{(3)}(n,:)\|_2}\right)$$

where $\boldsymbol{Y}_S^{(3)}(n,:)$ and $\hat{\boldsymbol{Y}}_S^{(3)}(n,:)$ represent the corresponding fibers of the ground-truth and the estimated super-resolution tensors, respectively. SAM measures the angles between the estimated and the ground-truth fibers of the SRI, and small SAMs correspond to good performance. *Relative dimensional global error (ERGAS)* [171] is also employed, which is defined as

$$\texttt{ERGAS} = 100d\sqrt{\frac{1}{IJK}\sum_{k=1}^{K}\frac{\|\underline{\hat{\boldsymbol{Y}}_S}(:,:,k) - \underline{\boldsymbol{Y}_S}(:,:,k)\|_F^2}{\mu_k^2}},$$

where $d = \frac{I_M}{I_H} = \frac{J_M}{J_H}$ and μ_k is the mean of the elements in $\underline{\boldsymbol{Y}_S}(:,:,k)$—and small ERGAS values are desired. In addition to the above quality measures, we also employ the *reconstruction Signal-to-Noise ratio (R-SNR)* criterion, i.e.,

$$\texttt{R-SNR} = 10\log_{10}\left(\frac{\sum_{k=1}^{K}\|\underline{\boldsymbol{Y}_S}(:,:,k)\|_F^2}{\sum_{k=1}^{K}\|\underline{\hat{\boldsymbol{Y}}_S}(:,:,k) - \underline{\boldsymbol{Y}_S}(:,:,k)\|_F^2}\right),$$

and high R-SNR values indicate good reconstruction performance.

3.5.1 Semi-Real Data Experiments

In this subsection, we test STEREO under the assumption that both $\boldsymbol{P}_M$ and $\boldsymbol{P}_H$ are known. A real hyperspectral image is used to act as the SRI in our simulations. This way, the 'ground-truth' SRI is known so that the performance can be easily measured. The corresponding HSI and MSI are degraded from this SRI following Wald's protocol [172] as described before. The degradation process from the SRI to the HSI is modeled as a combination of spatial blurring by a 9×9 Gaussian kernel and downsampling the blurred image by a factor of $d = 4$ along the two spatial directions.

The first experiment is performed using the dataset that is a subscene of SALINAS HSI from the AVIRIS platform. This scene describes a field that consists of 6 different agricultural products. The image is measured at 224 spectral bands. After removing 20 bands corrupted by water absorption we obtain an 'SRI' of 80×84 pixels with 204 bands, i.e. $\underline{\boldsymbol{Y}}_S \in \mathbb{R}^{80\times84\times204}$. Then, $\underline{\boldsymbol{Y}}_H \in \mathbb{R}^{20\times21\times204}$ is produced through the aforementioned spatial degradation, and

$\underline{\boldsymbol{Y}}_M \in \mathbb{R}^{80\times 84\times 6}$ is produced through LANDSAT spectral degradation. The rank used in the tensor decomposition is $F = 100$. For the matrix factorization methods, the number of endmembers (model rank) is set to be $R = 6$—which is equal to the ground-truth number of materials. For the `FUMI` algorithm, in order to satisfy the unit-box constraint that the algorithm makes use of (see details in [176]), the HSI and MSI pixels are normalized by the maximum entry of the MSI.

Table 3.5 shows the performance of the algorithms. It is clear that `STEREO` significantly outperforms the benchmarks. Particularly, in terms of `R-SNR`, `STEREO` outperforms `FUMI`, which admits the best `R-SNR` among the baselines, by 10 dB. Furthermore, the execution time of the proposed algorithms is very low (~ 1.3 sec.—similar as most of the matrix based methods), which makes the tensor based approach rather appealing. We also visualize one band of the estimated SRI in Fig. 3.3. One can see that the image produced by `STEREO` is indeed much more visually closer to the ground-truth SRI.

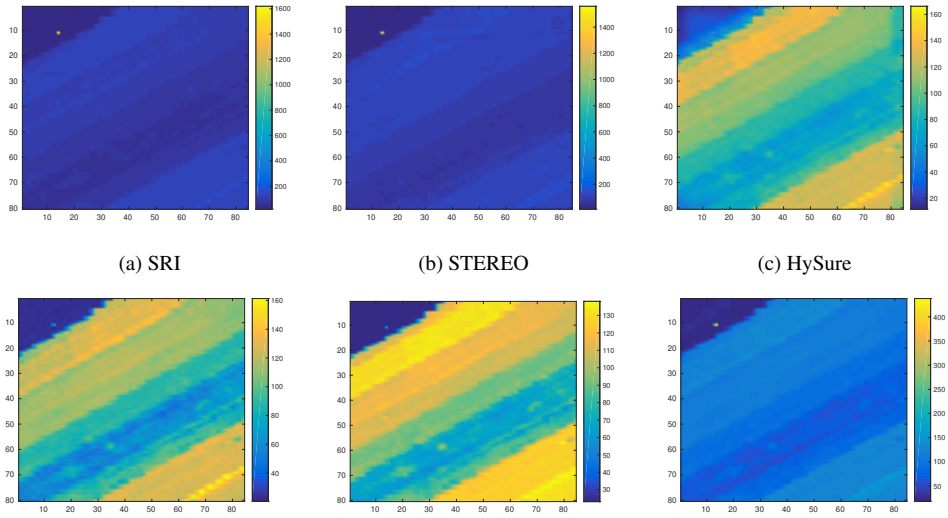
Table 3.5: SALINAS scene

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
`STEREO`	**38.62**	**0.9829**	0.5495	**1.3844**	1.3
`FUSE`	28.71	0.9174	**0.4234**	5.7135	**0.07**
`FUSE-Sparse`	28.71	0.9173	**0.4234**	5.7135	69.7
`FUMI`	29.40	0.9126	0.7975	6.3527	1.56
`HySure`	26.86	0.8981	1.5209	6.4187	1.6
`CNMF`	25.48	0.9013	1.3225	6.3787	1.7

The second experiment tests the super-resolution methods under scenarios where the degradation models are noisy. The Cuprite HSI downloaded from the AVIRIS platform is used to act as the SRI. The employed subimage has 187 bands (after removing bands corrupted by water absorption) and describes a spatial area containing 512×614 pixels, i.e., $\underline{\boldsymbol{Y}}_S \in \mathbb{R}^{512\times 614\times 187}$. The HSI and MSI are generated as follows:

$$\underline{\boldsymbol{Y}}_H = \underline{\boldsymbol{Y}}_S \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 + \underline{\boldsymbol{N}}_H$$
$$\underline{\boldsymbol{Y}}_M = \underline{\boldsymbol{Y}}_S \times_3 \boldsymbol{P}_M + \underline{\boldsymbol{N}}_M,$$

where $\underline{\boldsymbol{N}}_H$ and $\underline{\boldsymbol{N}}_M$ are additive white Gaussian noise. The degradation operators $\boldsymbol{P}_H = \boldsymbol{P}_2 \otimes \boldsymbol{P}_1$ and $\boldsymbol{P}_M$ are created as before, leading to $\underline{\boldsymbol{Y}}_H \in \mathbb{R}^{128\times 152\times 187}$ and $\underline{\boldsymbol{Y}}_M \in \mathbb{R}^{512\times 614\times 6}$,



(a) SRI (b) STEREO (c) HySure

(d) CNMF (e) FUMI (f) FUSE

Figure 3.3: SALINAS Reconstruction, 1442nm band

respectively. The signal-to-noise ratio (SNR) is defined as:

$$\texttt{SNR} = 10\log_{10}\left(\frac{\sum_{k=1}^{K}\|\underline{\boldsymbol{Y}}(:,:,k)\|_F^2}{\sum_{k=1}^{K}\|\underline{\boldsymbol{N}}(:,:,k)\|_F^2}\right),$$

where $(\underline{\boldsymbol{Y}},\ \underline{\boldsymbol{N}})$ stands either for the pair $(\underline{\boldsymbol{Y}}_H,\ \underline{\boldsymbol{N}}_H)$ or $(\underline{\boldsymbol{Y}}_M,\ \underline{\boldsymbol{N}}_M)$. The algorithms are examined under different SNRs. In all cases, the SNRs of the HSI and MSI are assumed to be the same. The rank used for tensor decomposition is chosen following Theorems 3.1-3.2 and adjusted according to the SNR of each scenario. Precisely, as the SNR varies from 50dB to 20dB the tensor rank changes from $F = 750$ to $F = 100$. The intuition is to use fewer canonical dimensions when the noise level is higher, so that the noise corruption can be better discounted. To be more precise, higher noise levels result in higher signal degradation. Choosing higher rank models will then result in noise fitting rather than signal fitting, which is the desirable in our case. The rank of the low-rank matrix models is set to be $R = 12$, which is determined by the number of materials in the images. Results are averaged over 10 Monte-Carlo simulations.

Fig. 3.4 shows the R−SNR performance of the methods under different noise levels. One can see that under high SNR scenarios the proposed STEREO algorithm exhibits the best performance. The matrix-based methods also work very well for the Cuprite data when the noise is almost

absent. However, when the SNR drops under 30dB, STEREO vastly outperforms the baselines—which shows the robustness of the proposed method to modeling mismatches. FUSE−Sparse fails to operate due to memory overflow. FUSE seems to be the most vulnerable under noise, and HySure works best among the baselines. The rest of the evaluation metrics are shown in Fig. 3.5, from which a similar conclusion can be drawn. In terms of the runtime performance, FUSE is the most efficient algorithm, since it only involves very simple procedures. Nevertheless, its accuracy performance is heavily affected by model mismatches in the degradation process, which is undesired in practice. Among the rest, the proposed approach admits the lowest execution time. Note that when the noise level increases, there is a decreasing trend in the runtime of the tensor methods. This happens because the rank reduced when the noise level increased, and lower ranks demand less computational time for the tensor approach.

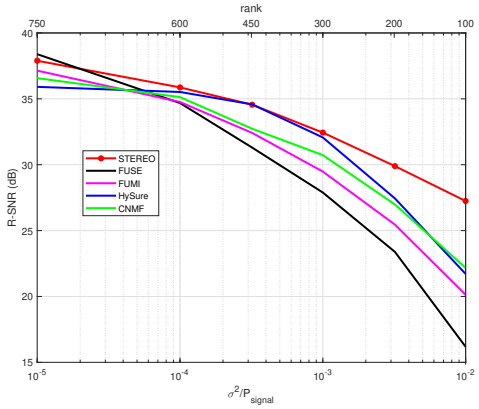
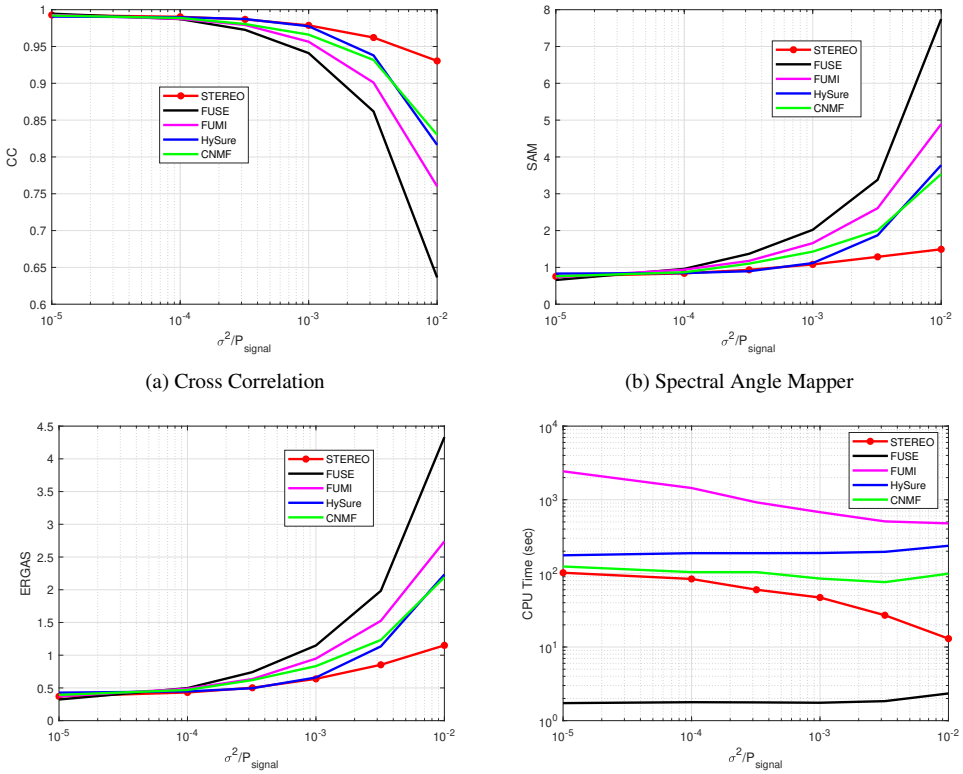


Figure 3.4: R−SNR of the algorithms on Cuprite under different noise levels.

Table 3.6 shows the performance of the algorithms when SNR=25dB. The tensor rank is set to be $F = 200$ in this case. STEREO produces the best results under all the evaluation metrics. Under this setup, HySure shows the best performance over all the evaluation metrics among the baseline algorithms, but it needs 3 times more runtime compared to that of STEREO. FUSE has the lowest runtime, but the R−SNR is 9dB worse relative to STEREO.

The algorithms are also tested on two more datasets. The first scene, namely, the Indian Pines, was again captured by AVIRIS and contains agriculture, forest and other natural perennial vegetation. The number of ground-truth materials is $R = 16$, and the pixels are measured at 200 bands (after removing water corrupted ones). We use $R = 16$ for all the baseline algorithms (except FUSE for which $R = 3$ is used, since it outputs particularly good results using this rank



(a) Cross Correlation

(b) Spectral Angle Mapper

(c) ERGAS

(d) CPU time

Figure 3.5: Reconstruction metrics for Cuprite

Table 3.6: Performance of the algorithms on the Cuprite data. SNR=25dB; "-" means "out of memory".

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**29.89**	**0.96216**	**1.2865**	**0.8533**	27
FUSE	23.38	0.8618	3.3793	1.9829	**1.8**
FUSE-Sparse	-	-	-	-	-
FUMI	25.45	0.9010	2.6078	1.5251	508.4
HySure	27.44	0.9379	1.8722	1.1345	196.5
CNMF	26.98	0.93170	2.0027	1.2320	75.5

for this dataset). The SRI in the experiment has 144×144 pixels, i.e., $\underline{\boldsymbol{Y}}_S \in \mathbb{R}^{144\times144\times200}$. The HSI and MSI are generated as before, leading to $\underline{\boldsymbol{Y}}_H \in \mathbb{R}^{36\times36\times200}$ and $\underline{\boldsymbol{Y}}_M \in \mathbb{R}^{144\times144\times6}$. Table 3.7 shows the performance when the SNR is 25 dB. The tensor rank is $F = 50$. Again, STEREO outperforms the baselines significantly.

Table 3.7: Performance of the algorithms for Indian Pines data. SNR=25dB.

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**25.80**	**0.8077**	**2.5217**	**1.3013**	1.8
FUSE	24.67	0.7469	2.8563	1.6665	**0.2**
FUSE-Sparse	24.67	0.7469	2.8563	1.6665	116.7
FUMI	23.54	0.7593	3.3931	1.8151	28.8
HySure	24.56	0.7710	2.8371	1.5938	12.1
CNMF	23.84	0.7321	3.0184	1.8227	4.2

The other scene is taken from Pavia University in Italy and was captured by the ROSIS sensor. The SRI, HSI, and MSI are with sizes of $608 \times 336 \times 103$, $152 \times 84 \times 103$ and $608 \times 336 \times 4$, respectively, in which we simulate a QuickBird-generated MSI. Table 3.8 shows the performance under SNR=25dB. The tensor rank is $F = 400$, and $R = 9$ is the ground-truth number of materials. One can see that STEREO shows superior performance in this simulation as before.

Table 3.8: Performance of the algorithms for Pavia University data. SNR=25dB; "-" means "out of memory".

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**22.50**	**0.9830**	**4.551**	**2.6016**	26.4
FUSE	21.09	0.9753	5.536	3.4284	**0.5**
FUSE-Sparse	-	-	-	-	-
FUMI	21.56	0.9779	5.1151	3.0908	644.2
HySure	21.18	0.9792	4.812	2.7934	82.5
CNMF	19.93	0.9723	5.0183	3.3947	19.2

Finally the proposed STEREO is examined under different choices of the main tuning parameters, namely, tensor rank F and λ. Figure 3.6 shows the R-SNR performance of STEREO under different choices of F. One can see that under different SNRs, there is always a wide range of F's (spanning several hundreds of consecutive integers) under which the proposed algorithm works reasonably well. Similar experiments are conducted to test the performance of STEREO under different choices of parameter λ. Figure 3.7 shows the achieved R-SNR of STEREO when λ varies form 0.01 to 100. The result shows that STEREO is quite insensitive to the choice of λ.

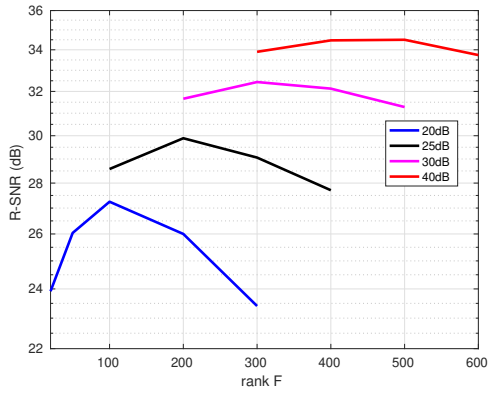


Figure 3.6: The obtained R−SNRs (dB) using STEREO under different SNRs and F's.

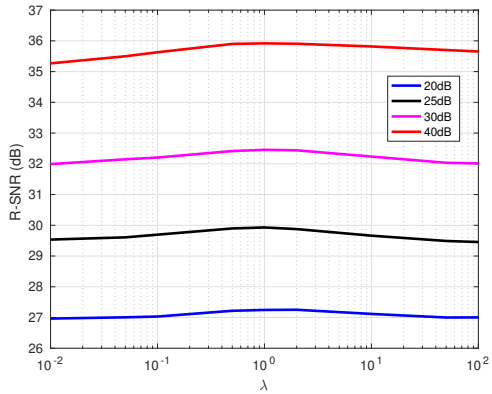


Figure 3.7: The obtained R−SNRs (dB) using STEREO under different SNRs and λ's.

3.5.2 Unknown Spatial Degradation Operator

In this subsection, we test our proposed Blind STEREO algorithm under the case where the spatial degradation model is unknown. The SRI used are the Indian Pines and Pavia University images as in the previous section. The HSI $\underline{\boldsymbol{Y}}_H$ is produced by $\underline{\boldsymbol{Y}}_S$ after 9×9 *Gaussian* blurring and downsampling and the MSI $\underline{\boldsymbol{Y}}_M$ is generated according to LANDSAT and QuickBird specifications, for Indian Pines and Pavia University image respectively.

We fist consider a case where the baseline algorithms falsely assume a 5×5 Gaussian blurring kernel instead of using the correct 9×9 Kernel. Among the baselines, HySure is able to estimate the degradation operators by assuming knowledge of the Kernel size and alignment

offset hyperparameters. The SNR of the degradation processes is 25dB. Table 3.9 shows the performance of the algorithms under this scenario using the Indian Pines image. The tensor rank is set to be $F = 50$. One can see that the proposed algorithm yields clearly better reconstruction performance under all the metrics. This shows the advantage of Blind STEREO—since it does not need to assume any prior knowledge on $\boldsymbol{P}_H$, the considered model mismatches do not affect its performance. Fig. 3.8 visualizes a band of the reconstructed super-resolution images by the algorithms. One can see that Blind STEREO gives visually more pleasing reconstruction relative to the baselines. The reconstruction performance in the Pavia University image is shown in Table 3.10, where the tensor rank is $F = 400$. One can see similar results there.

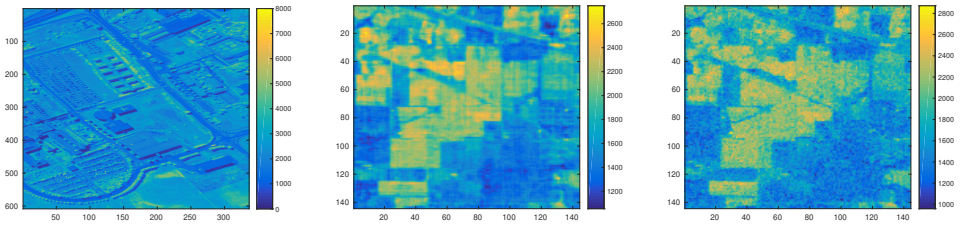
Table 3.9: Performance of the algorithms on the Indian Pines data under kernel size mismatch

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**25.53**	**0.7949**	**2.5831**	**1.3491**	1
FUSE	24.66	0.7447	2.8570	1.6632	**0.18**
FUSE-Sparse	24.66	0.7447	2.8570	1.6632	118.3
FUMI	23.03	0.7020	3.6744	2.1357	66.7
HySure	24.64	0.7853	2.7724	1.5255	12.4
CNMF	24.5	0.7254	3.1102	1.8903	3.2

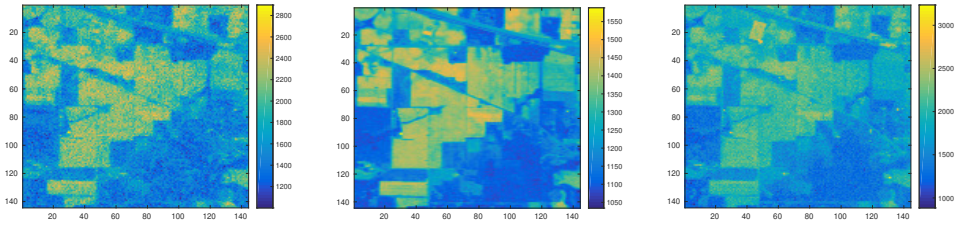
Table 3.10: Performance of the algorithms for Pavia University data under kernel size mismatch

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**22.36**	**0.9824**	**4.5997**	**2.6229**	26
FUSE	20.83	0.97347	5.4552	3.4906	**0.5**
FUMI	21.16	0.9763	5.045	3.1508	593.3
HySure	20.68	0.9773	4.868	2.902	82.4
CNMF	19.93	0.9727	5.0695	3.2938	19.3

We further consider another scenario where the baseline algorithms correctly assume a 9×9 Gaussian kernel, but the assumed blurring kernel is applied to an area which is misaligned with the ground-truth blurring area by 2 pixels in both of spatial dimensions. Note that such misalignment could easily happen in practice. The results of the second scenario are presented in Tables 3.11 and 3.12. Fig. 3.9 illustrates the reconstruction performance of Pavia University image at a selected band. Again, one can see that Blind STEREO clearly outperforms the benchmarking algorithms.

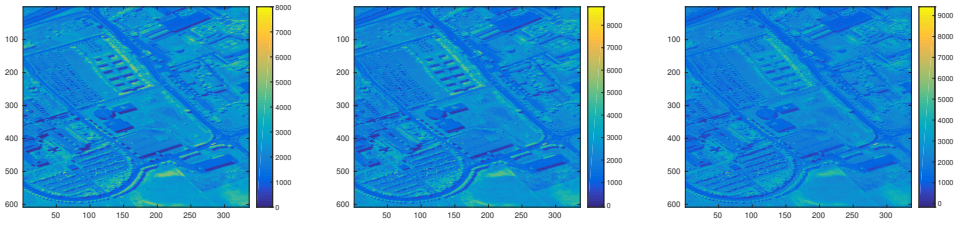


(a) SRI (b) STEREO (c) HySure

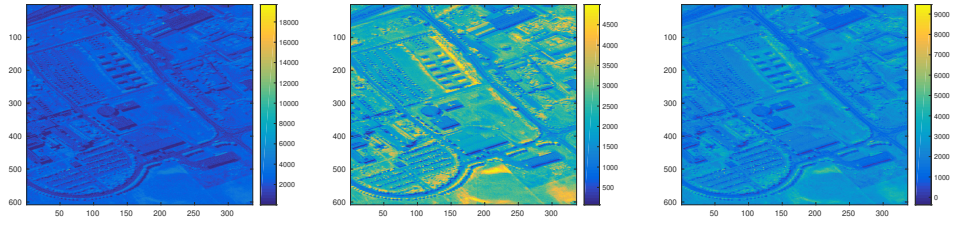


(d) CNMF (e) FUMI (f) FUSE

Figure 3.8: Indian Pines Reconstruction, 1422nm band



(a) SRI (b) STEREO (c) HySure



(d) CNMF (e) FUMI (f) FUSE

Figure 3.9: Pavia University Reconstruction, 858nm band

Table 3.11: Performance of the algorithms on the Indian Pines data under sampling offset mismatch

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**25.81**	**0.8198**	**2.5458**	**1.2788**	1.5
FUSE	23.90	0.7273	3.0148	1.7390	**0.14**
FUSE-Sparse	23.90	0.7273	3.0148	1.7390	118
FUMI	23.45	0.6782	3.8075	2.2039	55.1
HySure	23.23	0.7474	3.2332	1.7655	12.4
CNMF	24.01	0.7339	2.9896	1.8287	5.6

Table 3.12: Performance of the algorithms for Pavia University data under sampling offset mismatch.

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
STEREO	**22.36**	**0.9824**	**4.5997**	**2.6229**	26
FUSE	15.84	0.9283	7.2734	5.2655	**0.5**
FUMI	16.44	0.9392	5.8355	4.6652	287.8
HySure	17.64	0.9571	6.4415	3.8048	82.4
CNMF	19.93	0.9723	5.0183	3.3947	19.2

3.5.3 Simulations with SCUBA

In this subsection, we test our proposed Blind STEREO and SCUBA under settings where the spatial degradation is unknown. The model for the spatial degradation, we use to generate the HSI from the SRI (but we assume to be unknown) is the process of blurring by a 7×7 Gaussian kernel and downsampling 1 out of every $4 \times 4 = 16$ pixels of the result. The noise variance is controlled so that the HSI has 15 dB signal to noise ratio (SNR) and the MSI 25 dB SNR. Among the baselines SCUBA is fully blind, even for non-separable kernels, blind STEREO can perform blind spatial reconstruction, assuming a separable kernel, HySure can approximately estimate the spatial response when given the kernel size and downsampling offset and the rest need complete knowledge of the spatial degradation operator.

The first set of experiments uses the Cuprite HSI. Table 4.1 and figure 3.10 show the performance of the algorithms averaged over 10 Monte Carlo simulations. The rank used for blind STEREO is $F = 150$ and the rank of the low rank matrix model is $R = 10$. SCUBA divides the HSI and MSI into 16 non-overlapping blocks and for each block $F = 45$ and $R = 3$.

The second set of experiments, shown in table 3.14 and figure 3.11, uses the Pavia University HSI. For blind STEREO we use $F = 300$ and $R = 9$; for SCUBA we cut the images into 16

pieces and use $F = 120$ and $R = 3$ for each.

Summarizing the results, SCUBA shows the best super-resolution performance, whereas the previously proposed blind STEREO comes second. The results are even more remarkable if one notes that SCUBA and blind STEREO work without knowing the spatial degradation, while HySure is given the kernel size and downsampling offset and FUSE, FUMI and CNMF assume perfect knowledge of the spatial degradation. As far as time is concerned, FUSE is the fastest but gives low quality results. SCUBA and blind STEREO are the second fastest and SCUBA can even be fully parallelized across sub-tensor blocks.

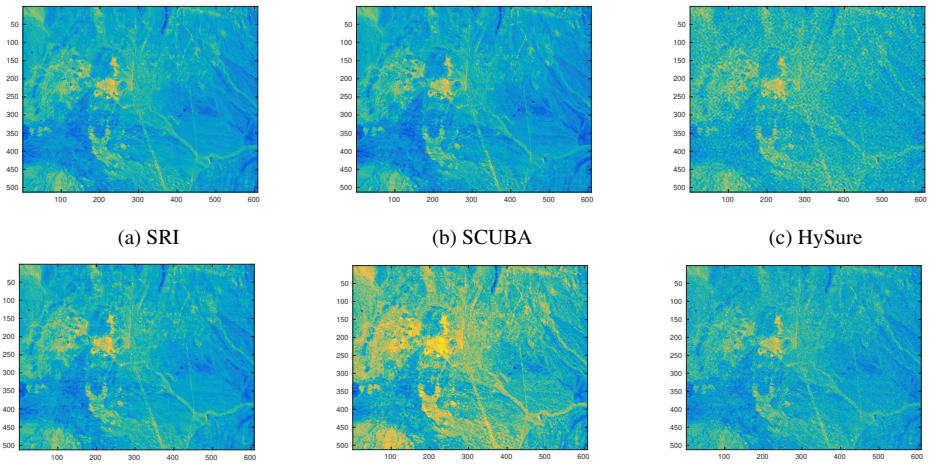
Overall, we see that SCUBA is computationally appealing, trivial to parallelize across blocks and fully blind in terms of the (possibly non-separable) spatial degradation – while retaining strong identifiability properties. Relative to blind STEREO, SCUBA further requires knowledge of (a bound on) the number of endmembers in each block, so the two are complementary to each other, in this sense.

Table 3.13: Performance of the algorithms in Cuprite Data.

Algorithm	**R-SNR**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
blind STEREO	27.88	0.9381	1.8004	1.1044	13.5
SCUBA	**29.06**	**0.9521**	**1.4695**	**0.9714**	15
FUSE	18.14	0.6952	6.4971	3.4517	**1.5**
FUMI	25.75	0.9069	2.4329	1.4176	89
HySure	24.20	0.8808	3.1095	1.6816	148
CNMF	22.97	0.8590	3.6957	1.9614	71.5

Table 3.14: Performance of the algorithms in Pavia University data

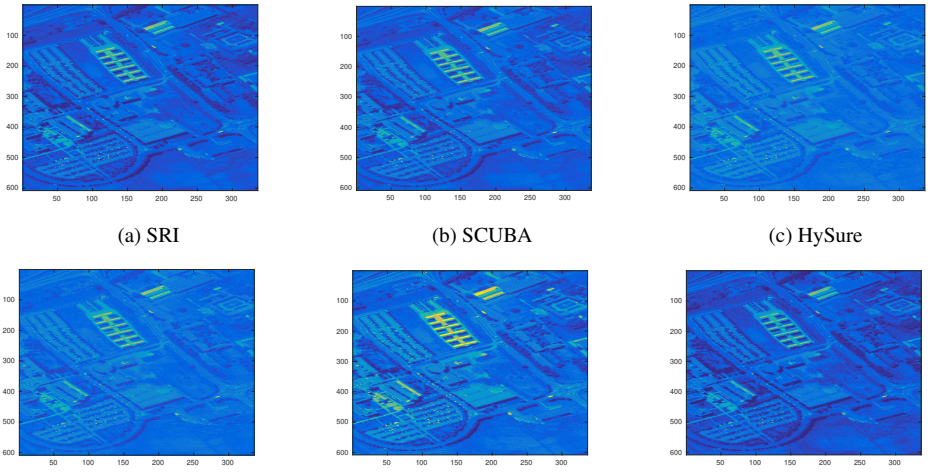
Algorithm	**NMSE**	**CC**	**SAM**	**ERGAS**	**runtime** (sec)
blind STEREO	20.39	0.9732	5.8279	3.3333	28
SCUBA	**22.84**	**0.9843**	**4.31**	**2.5624**	20.5
FUSE	19.43	0.96648	6.9954	3.9471	**0.6**
FUMI	22.01	0.9811	4.37	2.7328	116
HySure	19.89	0.9723	5.7094	3.1862	85
CNMF	18.43	0.9656	6.3049	3.9316	20



(a) SRI (b) SCUBA (c) HySure

(d) blind STEREO (e) FUMI (f) CNMF

Figure 3.10: Cuprite Reconstruction, 966nm band



(a) SRI (b) SCUBA (c) HySure

(d) blind STEREO (e) FUMI (f) CNMF

Figure 3.11: Pavia University Reconstruction, 554nm band

3.6 Conclusion

This chapter introduced a coupled tensor factorization and a hybrid tensor-matrix framework to tackle the hyperspectral super-resolution problem. Compared to the existing matrix-based approaches, the proposed methods show an array of theoretical advantages as well as more promising simulation results. Both methods are provably identifiable can easily accommodate scenarios where the spatial degradation operator is unknown or inaccurately estimated, which is usually the case in practice—without losing identifiability of the SRI. Extensive simulations using a variety of real-world hyperspectral images were conducted to evaluate the performance of our novel schemes.

Chapter 4

Tensor Completion from Regular Sub-Nyquist samples

In this chapter, we study regular sampling and reconstruction of three- or higher-dimensional signals (tensors)—or *tensor sampling* in short. Tensor signals naturally arise in a large number of areas such as machine learning and data analytics [125], signal processing and communications [55], image processing and remote sensing [80, 84, 85], medical imaging [39], genomics [70], chemometrics [148], just to name a few. Hence, considering sampling and reconstruction of tensor signals is of broad interest. The problem is challenging, since various tensor signals are neither bandlimited nor sparse or low-rank matrices (via 'unfolding')—and thus existing sampling techniques are not always applicable. We show that reconstructing a tensor signal from regular samples is feasible. Under the proposed framework, the sample complexity is determined by the tensor rank—rather than the signal bandwidth. This result offers new perspectives for designing practical *regular* sampling patterns and systems for signals that are naturally tensors, e.g., images and video. For a concrete application, we show that functional magnetic resonance imaging (fMRI) acceleration is a tensor sampling problem, and design practical sampling schemes and an algorithmic framework to handle it. Numerical results show that our tensor sampling strategy accelerates the fMRI sampling process significantly without sacrificing reconstruction accuracy. Part of this Chapter is published in [81, 88].

4.1 Prior Art

Sampling and reconstruction of signals is a fundamental problem in signal processing. In the first half of the 20th century, Whittaker, Nyquist, Kotelnikov and Shannon [96, 120, 139, 178] laid the foundation of *sampling theory*. It guarantees perfect reconstruction of a signal from uniformly spaced samples, if sampling is performed at a rate of at least twice the maximum frequency present in the signal. The Shannon-Nyquist theorem applies to both continuous and discrete signals. It capitalizes on the band-limitedness property and is the first, and one of the very few results, that allow perfect reconstruction of a signal under a uniform, or more generally, regular sampling process. The challenge is that applying Shannon-Nyquist sampling to wideband signals requires very high sampling rates—which entail high prohibitive complexity, size, and power consumption. Sub-Nyquist sampling and reconstruction strategies were studied as early as the late 60's for multiband signals [102], and the interest continued [165] until more recently, when sparsity came into play [31, 32, 49, 115].

In the late 2000's *compressive sensing* (CS) [31,32,49] emerged, enabling reconstruction from a set of measurements sampled or compressed below the Nyquist rate. CS works under two basic premises: the signal of interest must have a sparse representation in a known transform domain; and the sampling pattern should be 'incoherent'. Under these assumptions, tractable algorithms are shown to recover the signal of interest. Compared to the Shannon-Nyquist sampling theorem, CS leverages signal sparsity, rather than bandlimitedness. This result is significant, since some wideband signals of practical interest are sparse in certain domains [18, 113]. On the downside, CS entails higher reconstrunction complexity than sinc function interpolation, and relies on incoherent/random sampling thus losing the simplicity of regular/uniform sampling. A few exceptions exist, e.g., [46, 65], but the results are quite restrictive in practice.

Following the ideas of CS, *low-rank matrix completion* (LRMC) techniques were proposed for reconstructing matrix signals from a set of samples [30, 75]. This line of research utilizes the *rank* of the matrix as complexity measure for sampling and has attracted significant attention, since it is related to a number of important applications such as recommender systems [95]. However, similar to CS, LRMC is based on incoherent sampling. Furthermore the reconstruction guarantees in both CS and LRMC are probabilistic, contrary to the Shannon-Nyquist theorem which deterministically guarantees signal reconstruction.

Naturally, CS and LRMC ideas have been extended to higher dimensional signals and in

particular tensors. The reconstruction of sampled tensor signals, known in the literature as *tensor completion*, has been studied in machine learning and computer vision [90, 108]. The majority of existing works [3, 56, 108, 168, 188] focus on the algorithmic aspects of tensor completion. There are a few that provide recovery guarantees [71, 190] but are based on random sampling schemes and/or LRMC ideas, which are not tailored to the tensor specifics. A recent work [11], studies identifiability conditions of low-rank tensor completion with generic matrix factors. The sampling procedure is not constrained to be random, unlike [71, 190], but checking the conditions is a combinatorial problem. The work that is closest to ours is [152], which offers reconstruction conditions when the tensor 'fibers' are sampled. However, the conditions are restrictive, since the rank is constrained to be lower than the fiber dimension, and a variety of other interesting types of regular tensor sampling have not been considered.

4.2 Tensor Sampling Mechanisms

The core of this chapter discusses the sampling and reconstruction of third-order tensors. The main claim is fundamental: roughly speaking, any third-order tensor that does not have very high rank can always be identified from a sufficient number of regular samples. The sampling is not constrained to follow a randomized or incoherent process. On the contrary, we focus on *regular* and highly structured schemes. Various regular sampling strategies are considered. They involve sampling whole slabs in different modes (slab sampling), certain fibers in a single or multiple modes (fiber sampling) and entries in a systematic manner (entry sampling). Exposition and development use *third-order* tensors, but all the techniques can be naturally extended to higher-order tensors in a conceptually straightforward way. Similar to the case of matrices, even if a tensor is high-rank in the strict mathematical sense, it can often be approximated using low rank, in which case it can be *approximately* recovered using the proposed sampling and reconstruction schemes, as we will see.

4.2.1 General Strategy and Insight

Let us consider the following general form of tensor sampling:

$$\boldsymbol{y} = \mathsf{Sample}(\underline{\boldsymbol{X}}),$$

where $\mathsf{Sample}(\cdot) : \mathbb{R}^{I\times J\times K} \to \mathbb{R}^L$ is a down-sampling operator with $L \ll IJK$. Our goal is to study under what conditions and sampling strategies, identifying $\underline{\boldsymbol{X}}$ from $\boldsymbol{y}$ is possible. This is an inverse problem like in CS [31, 32, 49] and LRMC [30, 75]. However, unlike in [30–32, 49, 56, 75], we do not consider random/incoherent down-sampling operators but highly structured ones—which model a plethora of engineering applications, are easier for practical system implementation and computationally more efficient.

Our work rests upon two basic ideas. The first utilizes the uniqueness property of the CPD. Recall that every tensor admits a CPD, and the CPD is essentially unique if the CP rank is not very large. The second exploits the relation between a sampled sub-tensor and the original tensor $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. If we sample $\mathcal{S}_r \subseteq \{1, \ldots, I\}$ rows, $\mathcal{S}_c \subseteq \{1, \ldots, J\}$ columns and $\mathcal{S}_f \subseteq \{1, \ldots, K\}$ fibers and form a sub-tensor $\underline{\boldsymbol{X}}(\mathcal{S}_r, \mathcal{S}_c, \mathcal{S}_f)$, then:

$$\underline{\boldsymbol{X}}(\mathcal{S}_r, \mathcal{S}_c, \mathcal{S}_f) = [\![\boldsymbol{A}(\mathcal{S}_r, :), \boldsymbol{B}(\mathcal{S}_c, :), \boldsymbol{C}(\mathcal{S}_f, :)]\!].$$

One key observation is that the above sub-tensor can be decomposed to a sum of rank-one terms of number equal to the rank of the original tensor. Furthermore, the latent factors share certain rows with the original latent factors. Intuitively, if $\mathrm{rank}(\underline{\boldsymbol{X}})$ is not huge, there is a good chance that the sub-tensor admits a unique CPD, and part of the information of $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ can be extracted from the sub-tensor. Hence, by judiciously sampling and constructing sub-tensors, it seems viable to recover the entire $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$, and thus reconstruct $\underline{\boldsymbol{X}}$. This is the main idea.

Despite this conceptual simplicity, however, fleshing out this task is nontrivial. First, when factoring the sub-tensors there are always permutation and scaling ambiguities—even if every sub-tensor admits unique CPD, identifiability of the whole tensor is not guaranteed. Thus the sampling mechanisms need to be carefully designed to address this issue. Second, balancing the sampling ratio with the ability to identify the original tensor is a key consideration and needs attentive thinking and design. In the remaining section, we propose a series of sampling mechanisms that take into consideration both design challenges. The considered sampling schemes are practical and motivated by real engineering applications, particularly in the field of medical imaging.

4.2.2 Slab sampling

First, we study the task of reconstructing a third-order tensor from slab samples, taken from two different modes. Recovering tensor signals from sampled slabs finds applications in fMRI acceleration [41, 116, 150] and image/video inpainting [24, 127]. However, there is no unified characterization for recoverability under regular sampling patterns, to our best knowledge. Let

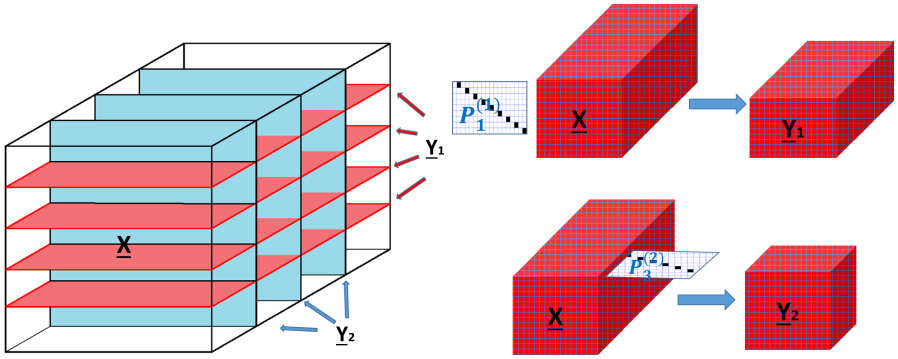


Figure 4.1: Tensor slab sampling paradigm.

$\underline{\boldsymbol{X}} \in \mathbb{F}^{I\times J\times K}$ be the original full tensor, which is not fully accessible or is subject to sampling. Instead we sample/observe a subset of slabs in one mode, e.g., horizontal slabs, $\mathcal{S}_r \subseteq \{1,\dots,I\}$, and a subset of slabs in a different mode, e.g. frontal slabs, $\mathcal{S}_f \subseteq \{1,\dots,K\}$. If $|\mathcal{S}_r| = I_1 \geq 2$ and $|\mathcal{S}_f| = K_2 \geq 2$, two separate sampled tensors are formed, i.e., $\underline{\boldsymbol{Y}}_1 \in \mathbb{F}^{I_1\times J\times K}$ and $\underline{\boldsymbol{Y}}_2 \in \mathbb{F}^{I\times J\times K_2}$, which represent the subset of observable horizontal and frontal slabs of $\underline{\boldsymbol{X}}$ respectively. Apparently, $\underline{\boldsymbol{Y}}_1$ can be written as the mode 1 multiplication of tensor $\underline{\boldsymbol{X}}$ with selection matrix $\boldsymbol{P}_1^{(1)} \in \mathbb{R}^{I_1\times I}$, i.e.

$$\underline{\boldsymbol{Y}}_1 = \underline{\boldsymbol{X}}\,(\mathcal{S}_h,:,:) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1^{(1)} \tag{4.1}$$

and $\underline{\boldsymbol{Y}}_2$ as a mode 3 multiplication with matrix $\boldsymbol{P}_3^{(2)} \in \mathbb{R}^{K_2\times K}$, i.e.

$$\underline{\boldsymbol{Y}}_2 = \underline{\boldsymbol{X}}\,(:,:,\mathcal{S}_f) = \underline{\boldsymbol{X}} \times_3 \boldsymbol{P}_3^{(2)} \tag{4.2}$$

The sampling matrices $\boldsymbol{P}_1^{(1)},\ \boldsymbol{P}_3^{(2)}$ perform slab selection in a single mode of $\underline{\boldsymbol{X}}$, thus $I_1 < I,\ K_2 < K$ (they are 'fat') and also have full row rank. A schematic illustration of the tensor slab sampling model is given in Fig. 4.1. Note that, $\boldsymbol{P}_1^{(1)},\ \boldsymbol{P}_3^{(2)}$ are not constrained to be randomly

drawn in our framework. On the contrary, the sampling process is allowed to be regular or highly structured, see Fig. 4.1. Assuming $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$, following (4.1), (4.2), the sub-tensors $\underline{\boldsymbol{Y}}_1,\ \underline{\boldsymbol{Y}}_2$ can be expressed in a PD form:

$$\underline{\boldsymbol{Y}}_1 = \left[\!\!\left[\boldsymbol{P}_1^{(1)}\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\right]\!\!\right] \tag{4.3a}$$

$$\underline{\boldsymbol{Y}}_2 = \left[\!\!\left[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3^{(2)}\boldsymbol{C}\right]\!\!\right] \tag{4.3b}$$

Using (4.3) identifiability of $\underline{\boldsymbol{X}}$ from $(\boldsymbol{Y}_1, \boldsymbol{Y}_2)$ can be established:

Theorem 4.1. *Let $\underline{\boldsymbol{X}} \in \mathbb{F}^{I\times J\times K}$ be the original tensor signal to recover, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ of rank F. Assume that $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are drawn from some joint absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{F}^{(I+J+K)F}$, and that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ satisfy the equations in* (4.3)*. Then, $\hat{\underline{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ recovers the ground-truth $\underline{\boldsymbol{X}}$ almost surely if one of the following conditions hold:*

1. $\min\left\{2^{\lfloor\log_2 I_1\rfloor+\lfloor\log_2 J\rfloor}, 2^{\lfloor\log_2 J\rfloor+\lfloor\log_2 K\rfloor},\ 2^{\lfloor\log_2 I_1\rfloor+\lfloor\log_2 K\rfloor}, 4JK_2\right\} \geq 4F$
2. $\min\left\{2^{\lfloor\log_2 I\rfloor+\lfloor\log_2 J\rfloor}, 2^{\lfloor\log_2 J\rfloor+\lfloor\log_2 K_2\rfloor},\ 2^{\lfloor\log_2 I\rfloor+\lfloor\log_2 K_2\rfloor}, 4I_1J\right\} \geq 4F,$

where $I_1, K_2 > 1$.

The proof is presented in Appendix C.1. The intuition is that if $\underline{\boldsymbol{Y}}_1$ or $\underline{\boldsymbol{Y}}_2$ admit a unique CPD, under Theorem 2.1, the factors $\boldsymbol{B},\ \boldsymbol{C}$ or $\boldsymbol{A},\ \boldsymbol{B}$ respectively can be identified. Then $\boldsymbol{A}$ or $\boldsymbol{C}$ are recovered from the other tensor, where $\boldsymbol{A}$ or $\boldsymbol{C}$ have been left uncompressed. Note that in slab sampling only one sub-tensor is required to admit a unique CPD. The reason is that identifying the latent factors of one sub-tensor, directly estimates two original latent factors. Then, the remaining factor can be obtained via solving a linear system of equations. Furthermore, permutation and scaling ambiguities are automatically resolved, since $\underline{\boldsymbol{Y}}_1,\ \underline{\boldsymbol{Y}}_2$ sample common rows of $\underline{\boldsymbol{X}}$. Overall reconstruction is performed as $\hat{\underline{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$. Deterministic conditions can also be derived, and this discussion is postponed to section 4.3.

The previous analysis can be easily extended to the case where slab sampling is performed in all 3 modes of the tensor.

4.2.3 Fiber sampling

Next, we consider the reconstruction of tensor $\underline{\boldsymbol{X}}$ from a subset of fibers, sampled along a single mode of the tensor. Fiber sampling is also of interest to a number of applications in chemometrics [3, 158] nuclear magnetic resonance spectroscopy [121] and fMRI acceleration (see Sec. 4.5). To make the discussion concrete, consider the scenario illustrated in Fig. 4.2, where $D = 3$ fiber patterns appear in the tensor sampling scheme. A pattern will be defined

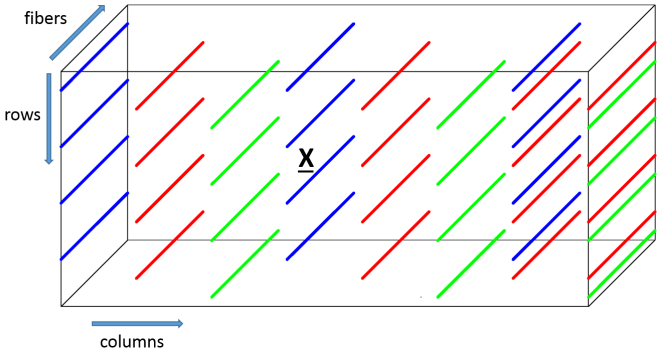


Figure 4.2: Tensor fiber sampling paradigm.

as a subset of rows $\mathcal{S}_r^{(d)} \subseteq \{1, \ldots, I\}$ and columns $\mathcal{S}_c^{(d)} \subseteq \{1, \ldots, J\}$ for which every point $\underline{\boldsymbol{X}}(i, j, k),\ i \in \mathcal{S}_r,\ j \in \mathcal{S}_c,\ k \in \{1, \ldots, K\}$ belongs to the pattern. In the illustrated scenario, each pattern (blue, d=1; red, d=2; green, d=3) samples fibers defined by the following subset of rows and columns: $\mathcal{S}_r^{(1)} = \{1, 4, 7, 10\},\ \mathcal{S}_c^{(1)} = \{1, 4, 7\},\ \mathcal{S}_r^{(2)} = \{2, 5, 8, 11\}, \mathcal{S}_c^{(2)} = \{2, 5, 7, 8\},\ \mathcal{S}_r^{(3)} = \{3, 6, 9, 12\}, \mathcal{S}_c^{(3)} = \{3, 6, 8\}$. Rearranging the order of the columns results in the model shown in Fig. 4.3.

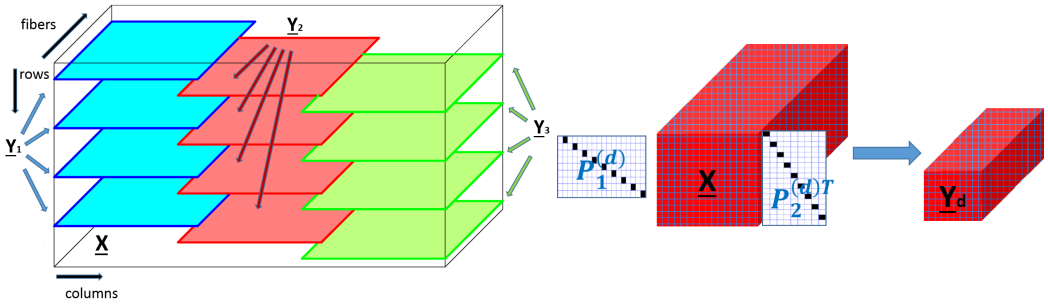


Figure 4.3: Fiber sampling model in a single mode

In the general case, the proposed fiber sampling framework entails each pattern forming a third-order tensor, i.e., $|\mathcal{S}_r^{(d)}|,\ |\mathcal{S}_c^{(d)}| \geq 2$ and that samples are taken from every row and

column of the tensor. The latter is a necessary condition for every factorization based completion approach, since completely unobserved slabs are impossible to recover. Furthermore, each pattern is required to sample from a common row or column with at least one more, thus creating an overlapping chain between patterns. The reason is that for pairwise mutually exclusive patterns, there exists a non-trivial scaling ambiguity, which cannot be determined. Formally the necessary sampling rules, for the proposed fiber sampling framework, are expressed as:

$$|\mathcal{S}_r^{(d)}|,\ |\mathcal{S}_c^{(d)}| \geq 2 \tag{4.4a}$$

$$\bigcup_{d=1}^{D} \mathcal{S}_r^{(d)} = \{1,\dots,I\},\ \bigcup_{d=1}^{D} \mathcal{S}_c^{(d)} = \{1,\dots,J\} \tag{4.4b}$$

$$\bigcup_{d'} \left\{ \mathcal{S}_c^{(d)} \cap \mathcal{S}_c^{(d')} \bigcup \mathcal{S}_r^{(d)} \cap \mathcal{S}_r^{(d')} \right\} \neq \emptyset,\ \forall\, d \in \{1,\dots,D\}\,, \tag{4.4c}$$

where $d \in \{1,\dots,D\}$, $d' \in \{1,\dots,D\} \setminus d$. The rules in 4.4 handle a plethora of sampling schemes. Specifically, each pattern is allowed to be equispaced, regular, random etc. This shows that reconstruction from regular samples is indeed doable. The sampling in Fig. 4.2, 4.3, for example, is regular and each pattern consists of equispaced rows and deterministically spaced columns.

Following similar analysis as in slab sampling, let $\underline{\boldsymbol{Y}}_d \in \mathbb{F}^{I_d \times J_d \times K}$ be the sampled subtensor, formed by pattern d. Also let $\boldsymbol{P}_1^{(d)} \in \mathbb{R}^{I_d \times I},\ \boldsymbol{P}_2^{(d)} \in \mathbb{R}^{J_d \times J}$ be the row and column selection matrices determining the d pattern. Then $\underline{\boldsymbol{Y}}_d$ is written as follows.

$$\underline{\boldsymbol{Y}}_d = \underline{\boldsymbol{X}}\left(\mathcal{S}_r^{(d)}, \mathcal{S}_c^{(d)}, :\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1^{(d)} \times_2 \boldsymbol{P}_2^{(d)} = \left[\!\!\left[\boldsymbol{P}_1^{(d)}\boldsymbol{A}, \boldsymbol{P}_2^{(d)}\boldsymbol{B}, \boldsymbol{C} \right]\!\!\right],\ d = 1,\dots,D \tag{4.5}$$

Using the equation in (4.5) we can establish generic identifiability of fiber sampling as:

Theorem 4.2. *Let $\underline{\boldsymbol{X}} \in \mathbb{F}^{I \times J \times K}$ be the original tensor signal, fiber sampled according to* (4.4)*, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ of rank F. Assume that $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are drawn from some joint absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{F}^{(I+J+K)F}$, and that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ satisfy the equations in* (4.5)*. Then, $\hat{\underline{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ recovers the ground-truth $\underline{\boldsymbol{X}}$ almost surely if:*

$$2^{\min_d \{\lfloor \log_2 I_d \rfloor + \lfloor \log_2 J_d \rfloor, \lfloor \log_2 J_d \rfloor + \lfloor \log_2 K \rfloor, \lfloor \log_2 I_d \rfloor + \lfloor \log_2 K \rfloor\}} \geq 4F$$

The proof is relegated to Appendix C.2. In contrast to the previous case of slab sampling, where identifiability of one sampled tensor $\underline{\boldsymbol{Y}}_i$ is enough, fiber sampling requires all $\underline{\boldsymbol{Y}}_i$'s to admit a unique CPD model—otherwise certain rows of $\boldsymbol{A},\ \boldsymbol{B}$ would be impossible to identify. The claim is simple and intuitive: The number of samples required to identify a fiber sampled tensor is proportional to the rank of the tensor.

Remark 4.1. Theorem 4.2 studies general tensors where factor $\boldsymbol{C}$ is not required to have full column rank, and thus $K < F$ can be easily handled. Fiber sampling and recovery of tensors with $\boldsymbol{C}$ having full column rank, is extensively studied in [152]. Compared to our work, the sampling strategy therein has to follow rules (4.4b), (4.4c), whereas (4.4a) can be relaxed. On the other hand, the results of this chapter are tailored to cases where the sampling process exhibits some regularity and $K < F$ is allowed. Note that $\boldsymbol{C}$ being full column rank, which is mandatory in [152], is a quite restrictive condition and prohibitive for several applications, e.g., fMRI acceleration as we will see next.

4.2.4 Entry sampling

So far we have discussed slab, fiber sampling of third-order tensors and provided conditions under which identifiability is guaranteed. In this subsection, we move a step further and study the more general problem of tensor reconstruction from a subset of entries, sampled in a regular fashion along the tensor. Entry sampling is another important sampling mechanism, which along with fiber sampling will prove very useful in accelerating the fMRI scan acquisition (see Sec. 4.5).

We are interested in cases where the sampling process can be viewed as a series of patterns. A pattern is defined, similarly to fiber sampling, as a subset of rows $\mathcal{S}_r^{(d)} \subseteq \{1, \dots, I\}$, columns $\mathcal{S}_c^{(d)} \subseteq \{1, \dots, J\}$ and fibers $\mathcal{S}_f^{(d)} \subseteq \{1, \dots, K\}$, for which every point $\underline{\boldsymbol{X}}(i, j, k),\ i \in \mathcal{S}_r, j \in \mathcal{S}_c, k \in \mathcal{S}_f$ belongs to the pattern. For example consider the scenario illustrated in Fig. 4.4. The number of patterns is $D = 3$ and $\mathcal{S}_r^{(1)} = \mathcal{S}_c^{(1)} = \{1, 3, 5, 7\},\ \mathcal{S}_f^{(1)} = \{1, 4\},\ \mathcal{S}_r^{(2)} = \mathcal{S}_c^{(2)} = \{2, 4, 6, 8\}, \mathcal{S}_f^{(2)} = \{2, 5\},\ \mathcal{S}_r^{(3)} = \mathcal{S}_c^{(3)} = \{3, 4, 5, 6\}, \mathcal{S}_f^{(3)} = \{3, 6\}$. In general, the proposed framework requires samples to be taken from all rows, columns and fibers of the tensor, thus $D \geq 2$ and each pattern should include 2 rows, 2 columns and 2 fibers at minimum. Furthermore, the overlap of the patterns should form a connected graph, i.e. if the nodes represent different patterns and the edges represent overlap, then this graph should be connected. The overlap between two patterns is required to involve at least 2 elements in one mode and 1 element in a

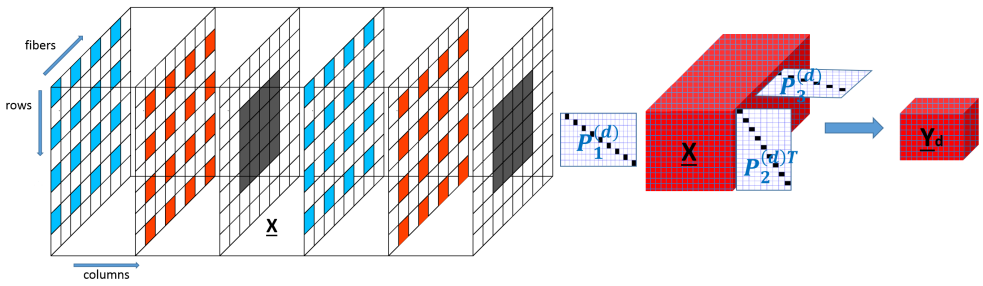


Figure 4.4: Tensor entry sampling paradigm. (Colored boxes represent sampled entries)

different mode. For example, a pair of patterns should sample 2 common rows and 1 common column. The latter is a necessary condition resulting from the inherent permutation and scaling ambiguity of the CPD. Formally the rules of entry sampling are:

$$|\mathcal{S}_r^{(d)}|,\ |\mathcal{S}_c^{(d)}|,\ |\mathcal{S}_f^{(d)}| \geq 2 \tag{4.6a}$$

$$\bigcup_{d=1}^{D} \mathcal{S}_r^{(d)} = \{1,\ldots,I\},\ \bigcup_{d=1}^{D} \mathcal{S}_c^{(d)} = \{1,\ldots,J\}, \bigcup_{d=1}^{D} \mathcal{S}_f^{(d)} = \{1,\ldots,K\} \tag{4.6b}$$

$$\forall\ d \in \{1,\ldots,D\}\,,\ \exists \text{ a pair of } m,\ m' \text{ such that}$$
$$\bigcup_{d'} \left\{ \mathcal{S}_m^{(d)} \cap \mathcal{S}_m^{(d')} \bigcap \mathcal{S}_{m'}^{(d)} \cap \mathcal{S}_{m'}^{(d')} \right\} \neq \emptyset \text{ and } |\mathcal{S}_m^{(d)} \cap \mathcal{S}_m^{(d')}| \geq 2, \tag{4.6c}$$

$$\mathcal{G} := \{\mathcal{V}, \mathcal{E}\} \text{ is connected,} \tag{4.6d}$$

where $d \in \{1,\ldots,D\}$, $d' \in \{1,\ldots,D\} \setminus d$, $m \in \{c,r,f\}\,,\ m' \in \{c,r,f\} \setminus m$, and $\mathcal{G}$ is an undirected graph, with $\mathcal{V}$ being the set of $D = |\mathcal{V}|$ nodes corresponding to different patterns, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of edges representing overlap between patterns.

Following similar analysis as in fiber sampling, let $\underline{\boldsymbol{Y}}_d \in \mathbb{F}^{I_d \times J_d \times K_d}$ be the sampled subtensor representation of pattern d. Also let $\boldsymbol{P}_1^{(d)} \in \mathbb{R}^{I_d \times I}$, $\boldsymbol{P}_2^{(d)} \in \mathbb{R}^{J_d \times J}$, $\boldsymbol{P}_3^{(d)} \in \mathbb{R}^{K_d \times K}$ be the row, column and fiber selection matrices determining pattern d. Then $\underline{\boldsymbol{Y}}_d$ is written as:

$$\begin{aligned} \underline{\boldsymbol{Y}}_d &= \underline{\boldsymbol{X}}\left(\mathcal{S}_r^{(d)}, \mathcal{S}_c^{(d)}, \mathcal{S}_f^{(d)}\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1^{(d)} \times_2 \boldsymbol{P}_2^{(d)} \times_3 \boldsymbol{P}_3^{(d)} \\ &= \left[\!\left[\boldsymbol{P}_1^{(d)}\boldsymbol{A}, \boldsymbol{P}_2^{(d)}\boldsymbol{B}, \boldsymbol{P}_3^{(d)}\boldsymbol{C} \right]\!\right] \qquad d = 1,\ldots,D \end{aligned} \tag{4.7}$$

The model in (4.7) is identifiable, under generic conditions presented in the following theorem.

Theorem 4.3. *Let $\underline{\boldsymbol{X}} \in \mathbb{F}^{I\times J\times K}$ be the original tensor signal, sampled according to* (4.6)*, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ of rank F. Assume that $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are drawn from some joint absolutely continuous distribution with respect to the Lebesgue measure in $\mathbb{F}^{(I+J+K)F}$, and that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ satisfy the equations in* (4.7)*. Then, $\hat{\underline{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ recovers the ground-truth $\underline{\boldsymbol{X}}$ almost surely if:*

$$2^{\min_d\{\lfloor \log_2 I_d\rfloor + \lfloor \log_2 J_d\rfloor, \lfloor \log_2 J_d\rfloor + \lfloor \log_2 K_d\rfloor, \lfloor \log_2 I_d\rfloor + \lfloor \log_2 K_d\rfloor\}} \geq 4F$$

The proof is presented in Appendix C.2. Similar to fiber sampling, identifiability of a tensor from entries, sampled as described in (4.6), is guaranteed, if all the sub-sampled tensors formed by the emerging patterns admit a unique CPD.

4.3 Deterministic Identifiability

The sampling mechanisms, discussed so far, can be realized as separate, yet coupled, sub-sampled versions of the original third-order tensor $\underline{\boldsymbol{X}}$. Identifiability of $\underline{\boldsymbol{X}}$, under various sampling mechanisms, was established by applying generic identifiability results on the CPD of the sub-tensors. However, the original tensor is also identifiable under a purely deterministic setting, i.e., the CPD factors of the tensor can be systematic (not necessarily drawn from an absolutely continuous distribution, which excludes measure-zero outcomes with probability one) and the conditions are deterministic. In the case of slab sampling we have the following theorem.

Theorem 4.4. *Let $\underline{\boldsymbol{X}} \in \mathbb{F}^{I\times J\times K}$ be the original tensor signal to recover, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ of rank F. Assume that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ satisfy the equations in* (4.3)*. Then, $\hat{\underline{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ recovers the ground-truth $\underline{\boldsymbol{X}}$ if $2F+2 \leq k_{\boldsymbol{P}_1^{(1)}\boldsymbol{A}^\star} + k_{\boldsymbol{B}^\star} + k_{\boldsymbol{C}^\star}$ and $\boldsymbol{B}^\star \odot \boldsymbol{P}_3^{(2)}\boldsymbol{C}^\star$ has full column rank, or if $2F+2 \leq k_{\boldsymbol{A}^\star} + k_{\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3^{(2)}\boldsymbol{C}^\star}$ and $\boldsymbol{B}^\star \odot \boldsymbol{P}_1^{(1)}\boldsymbol{A}^\star$ has full column rank.*

When fiber or entry sampling is employed, we have:

Theorem 4.5. *Let $\underline{\boldsymbol{X}} \in \mathbb{F}^{I\times J\times K}$ be the original tensor signal, fiber or entry sampled according to* (4.4) *or* (4.6) *respectively. Also let $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ denote the rank-F CPD of $\underline{\boldsymbol{X}}$. Assume*

that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ *have no repeated entries and satisfy the equations in* (4.5)*,* (4.7)*, according to the sampling mechanism. Then,* $\hat{\underline{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}^\star(i,f)\boldsymbol{B}^\star(j,f)\boldsymbol{C}^\star(k,f)$ *recovers the ground-truth* $\underline{\boldsymbol{X}}$ *if* $2F + 2 \leq \min_d \left\{ k_{\boldsymbol{P}_1^{(d)}\boldsymbol{A}^\star} + k_{\boldsymbol{P}_2^{(d)}\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3^{(d)}\boldsymbol{C}^\star} \right\}$*, where* $\boldsymbol{P}_3^{(d)} = \boldsymbol{I}$ *for fiber sampling.*

Proof of both theorems is presented in Appendix C.3.

Remark 4.2. Theorems 4.1-4.5 establish identifiability of third order tensors from a number of regular samples, in the sense that there is a single low-rank tensor completion that is consistent with the given samples. In simple words, factors $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ that solve equations (4.3), (4.5), (4.7), for slab, fiber and entry sampling respectively, and satisfy the conditions of Theorems 4.1-4.5 recover the original tensor. The caveat is that solving equations (4.3), (4.5), (4.7) to optimality is not an easy task. It involves computing the CPD of full sub-tensors, formed from the regular samples, which is NP-hard in general. However, there exist algebraic algorithms that solve the CPD of a tensor with known rank in polynomial time [48, 136], under conditions that are stricter than those for uniqueness [48]. Therefore, combining our conditions with those on algebraic computation of the CPD in equations (4.3), (4.5), (4.7) yields guaranteed recovery of tensor $\underline{\boldsymbol{X}}$ in polynomial time. Furthermore, there is a variety of advanced optimization algorithms, which are empirically effective in computing the CPD of a tensor. To summarize, although the discussed conditions focus on identifiability of the tensor signal and actual recovery is NP-hard in general, there exist algorithms that perform the recovery task in polynomial time under more restrictive conditions, and experience has shown that the more advanced optimization-based algorithms usually work well enough in practice. Computational and algorithmic aspects of the proposed framework are thoroughly discussed in section 4.6.

4.4 Further discussion and Insights

The implication of Theorems 4.1- 4.5 is significant and intuitive. Identifiability of $\underline{\boldsymbol{X}}$ is based on two basic principles: identifiability of the factors of the sub-sampled tensors and ability to reconcile for the permutation and scaling ambiguities. The first is a property of both the signal of interest and the sampling mechanism. In particular the rank of the tensor signal, along with Kruskal or generic conditions on the factors determine the number of samples required to identify the original tensor. Hence, there is a clear correlation between the rank of the tensor and the number of samples needed—higher ranks require higher number of samples. Note that

the number and structure of samples varies according to the applied sampling mechanism. The second principle is a necessary property of the sampling mechanism. Although slab sampling automatically handles permutation and scaling ambiguities, fiber and entry sampling schemes have to be carefully designed to satisfy (4.4) or (4.6) and eliminate permutation and scaling mismatches.

In a nutshell, one can learn the rows of factors $\boldsymbol{A},\ \boldsymbol{B},\ \boldsymbol{C}$ from the sub-tensors, up to column permutation and scaling and resolve the mismatches using common information between the sub-tensors. Then reconstruction of the original tensor is attained as $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$.

To build some further intuition on the theoretical conditions, consider the following example. Let $\underline{\boldsymbol{X}} \in \mathbb{F}^{512\times512\times512}$ be the tensor with CP rank $F = 1000$ which is subject to sampling. First we sample the I_1 equispaced horizontal slabs and K_2 equispaced frontal slabs. Following Theorem 4.1, identifiability of $\underline{\boldsymbol{X}}$ is guaranteed if we sample at least $I_1 = 8$ horizontal slabs and $K_2 = 2$ frontal slabs and vise versa. This results in sampling ratio $r = \frac{\#\text{observed entries}}{\#\text{total entries}} = 0.019$. Next we sample fibers of the tensor in a regular fashion, similarly to Fig. 4.2. According to Theorem 4.2, $33,216$ fibers are sufficient to identify the original tensor, which gives sampling ratio $r = 0.13$. Finally, entries are sampled in a regular fashion, as shown in Fig. 4.4. The total number of entries required to identify the original tensor is $2,870,336$, according to Theorem 4.3, which results in sampling ratio $r = 0.021$. Note that for smaller rank, e.g, $F = 250$ the total number of samples can be significantly reduced, giving sampling ratio $r_{slab} = 0.008$, $r_{fiber} = 0.064$, $r_{entry} = 0.0097$.

Another important question is how tight our conditions are, with respect to the degrees of freedom of the low rank CPD model. To facilitate the analysis we will assume that I is a power of 2, $I = J = K$ and the sampling is symmetric in the sampled modes, i.e., $\underline{\boldsymbol{Y}}_d,\ d = 1 \dots D$, are of same size. Then the degrees of freedom, due to the low rank CPD model are $3IF - 2F$ and the number of equations is equal to rI^3, where r is the previously defined sampling ratio. Therefore, the necessary (is not sufficient to guarantee identifiability) equations versus degrees of freedom bound yields:

$$3IF - 2F \leq rI^3 \Leftrightarrow r \geq \frac{3F}{I^2} - \frac{2F}{I^3} \tag{4.8}$$

We study each sampling mechanism separately:

- Slab sampling: The number of observed entries is $I_1I^2+(I-I_1)I_1I$, so $r = \frac{I_1I^2+(I-I_1)I_1I}{I^3}$.

The conditions of Theorem 4.1 yield $I_1 I \geq 4F$ and is easy to show that this condition is equivalent to:

$$r \geq \frac{8F}{I^2} - \frac{16F^2}{I^4},$$

which is same order of magnitude with (4.8).

- Fiber sampling: The number of observed entries is approximately $I_1 I^2 + I^2$, so $r = \frac{I_1+1}{I}$. Then the conditions of Theorem 4.2 boil down to $I_1^2 \geq 4F$ which is equivalent to:

$$r \geq \frac{2\sqrt{F}+1}{I}.$$

This bound is stricter compared to (4.8).

- Entry sampling: The number of observed entries is approximately $I_1^2 I + 3I^2$, so $r = \frac{I_1^2+3I}{I^2}$. The conditions of Theorem 4.3 yield $I_1^2 \geq 4F$, which is equivalent to:

$$r \geq \frac{4F}{I^2} + \frac{3}{I}.$$

In that case the necessary and the sufficient bounds are relatively close.

The previous analysis demonstrated that in the case of slab and entry sampling the sufficient condition for tensor identifiability from regular samples is relatively close to the necessary condition given by the degrees of freedom. In case of fiber sampling there is a non-negligible gap between the sufficient and naive necessary condition.

4.5 Application to parallel fMRI acceleration

Interestingly, the previously described sampling mechanisms find application in accelerating fMRI scan acquisition. fMRI is used to measure brain activity associated with changes in blood oxygen levels. MRI acquisitions typically use a set of coils (sensors), that in parallel collect a series of frames focusing on different parts of the brain. In fMRI, the three-dimensional (3D) volume covering the whole brain is typically acquired using multiple two-dimensional (2D) slices. These are discrete-space signals, sampled along a particular trajectory in the k-space, which is a 2-D frequency domain ($k_x,\ k_y$), for each brain slice. Therefore an fMRI scan, can be represented by a five-way array with coil, $k_x,\ k_y$, slice and time (frame) dimensions.

Acquiring high spatial resolution fMRI is challenging due to time restrictions. On the one hand, the k-space sampling has to follow the Shannon-Nyquist theorem to avoid artifacts, when inverse Fourier transform is used for reconstruction. On the other hand sampling at a Nyquist rate leads to prolonged scan acquisition time for each frame, which is prohibitive for high temporal resolution, required in fMRI and neuroscience research. The objective is therefore two-fold: Accelerate the scanning process and capture fast brain activity changes. Since the scan acquisition time is proportional to the number of k-space samples, ongoing efforts focus on sampling part of the k-space (k_y frequencies) of each slice and/or measuring the k-space of only a subset of slices. The majority of work is mainly proposed for MRI scans. Classic methods use learning and calibration type techniques [8, 61, 110], while others employ the CS framework [79, 110, 122] or LRMC [42, 79, 106, 142] to perform the reconstruction.

While MRI offers significant freedom in designing the k-space sampling trajectories for each frame, fMRI acquisition is more restrictive. Specifically, fMRI is performed using a special fast imaging acquisition, called echo planar imaging, that is practically only used with equispaced sub-sampling patterns due to restrictions associated with magnetic field inhomogenities and Eddy currents [52]. In simple words, the k_y frequencies sampled for each frame have to be equispaced and all coils need to measure the same frequencies. For example the sampling scheme illustrated in Fig. 4.5 is typical in fMRI and performs 3-fold acceleration. In general, a fully sampled

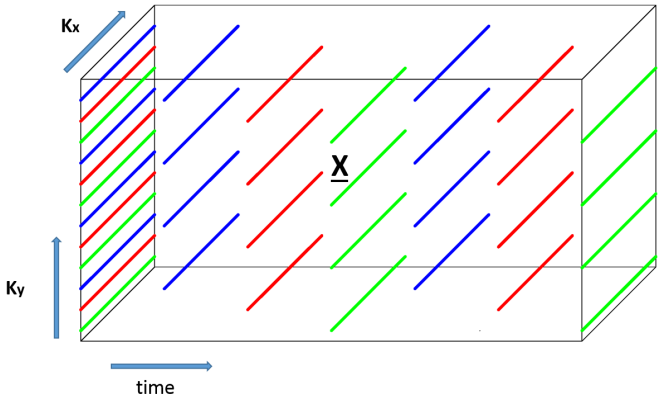


Figure 4.5: Single-slice fMRI sampling at each coil.

scan is acquired first, which is beneficial for calibration purposes. Then n-fold acceleration is achieved by sampling $1/n$ of equispaced k_y frequencies. The frequencies to sample for each frame can be the same for the whole procedure or can be circularly shifted as in Fig. 4.5. For the

benefit of our method we propose to circularly shift between n equispaced set of frequencies in order to capture the temporal behavior of the brain accurately. Note that this circular shift between the equispaced frequencies along with the first fully sampled scan guarantee that the rules in (4.4) are satisfied.

Sampling the k_y dimension is one way to accelerate the fMRI scanning process. Another idea that is being used is to observe the k-space of only a subset of slices at each time slot. In this chapter we propose to combine these two ideas to further reduce scanning time. Specifically at each time instance sub-sampled k-space measurements are acquired for only a subset of slices, instead of the complete set. To design a sampling mechanism that fits our tensor models and fMRI constraints, we first need to acquire a fully sampled scan for every slice. Then for each frame $1/\rho$ of equispaced k_y frequencies is observed for $1/s$ of the brain slices in a circular fashion, so that at the (ρs)-th frame we have measured every frequency for every slice. This results in (ρs)-fold acceleration. Fig. 4.6 illustrates an fMRI acceleration technique, where $\rho, s = 2$. Again, the first fully sampled scan and the circular sampling procedure guarantee that the rules in (4.6) are satisfied.

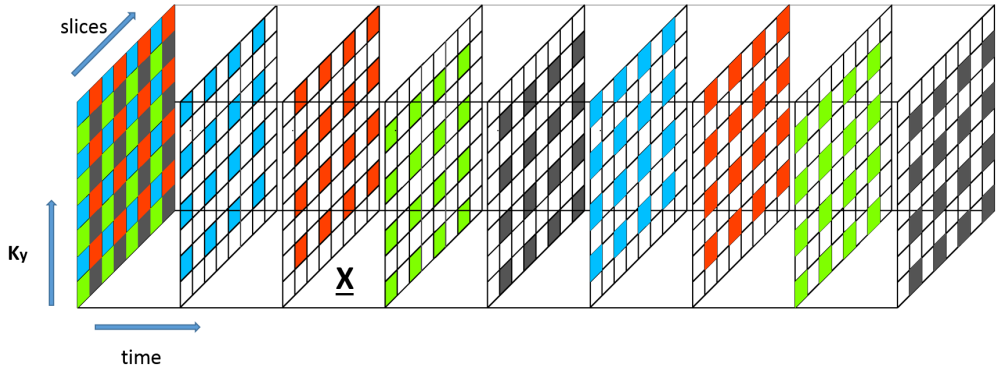


Figure 4.6: Multi-slice fMRI sampling at each coil.

Note that the two aforementioned sampling procedures can be tricky for classic techniques. On the one hand, calibration-based techniques such as GRAPPA [61] are linear and suffer from noise amplification at high acceleration rates. On the other hand, CS and LRMC schemes have difficulties in operating with regular samples, since their success rests upon incoherent sampling.

On the contrary, the proposed tensor sampling and reconstruction framework is exactly designed to handle these highly structured and constrained sampling schemes used in fMRI acquisitions. In particular, the single-slice fMRI acceleration task, as illustrated in Fig. 4.5,

can be cast as a tensor fiber sampling mechanism, analyzed in subsection 4.2.3. As mentioned earlier the raw fMRI scan is originally a five-way array and thus each slice is a four-way array. Although the previous analysis could be easily extended to tensors of order higher than three, we choose to work with third-order ones. Specifically the k-space is processed in a single dimension (mode) by concatenating k_x and k_y. The reason is that the relation between k_x and k_y is often hard to be captured by a multilinear tensor model. As a result one fMRI slice is modeled as a third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{C}^{I \times J \times K}$, where $I = m_x m_y$ with $m_x,\ m_y$ representing the number of frequencies in $k_x,\ k_y$ space respectively, J represents the total number of frames (time slots) and K the number of coils. Following the analysis of subsection 4.2.3 we have the following result.

Proposition 4.1. *Let $\underline{\boldsymbol{X}} \in \mathbb{C}^{I \times J \times K}$ be the a single-slice fMRI tensor with rank F, modeled as previously explained. Under the assumptions of Theorem 4.2, n-fold acceleration can be achieved if $n \leq \min\left\{\sqrt{\frac{IJ}{16F}}, \frac{JK}{16F}, \frac{IK}{16F}\right\}$.*

Similarly, the proposed multi-slice fMRI acceleration scheme, which performs joint k-space and slice sampling, is cast as an entry tensor sampling procedure, introduced in 4.2.4. To do so, the k-space is considered as a single mode, as before, and we also concatenate coils and slices in one dimension. The resulting third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{C}^{I \times J \times K}$ has the k-space in the first mode, i.e, $I = m_x m_y$, J represents the total number of frames and the third mode includes the concatenation of coils and slices, i.e., $K = m_s m_c$ with $m_s,\ m_c$ being the number of slices and coils respectively. Following the analysis of subsection 4.2.4 we have:

Proposition 4.2. *Let $\underline{\boldsymbol{X}} \in \mathbb{C}^{I \times J \times K}$ be the a multi-slice fMRI tensor with rank F, modeled as previously explained. Under the assumptions of Theorem 4.3, (ρs)-fold acceleration can be achieved if $\rho s \leq \frac{1}{16F} \min\left\{IK, \frac{JK}{s}, \frac{IJ}{\rho}\right\}$.*

We should mention that tensor approaches have also been proposed in medical imaging [16, 43, 66, 111, 114, 180]. The work in [114], for example, uses a tensor model to approach the MRI sampling and reconstruction problem in an on-line fashion. However, [114] works under different sampling schemes, which are not regular and appropriate for fMRI, and identifiability guarantees are not discussed. Moreover, a tensor model is also used in [180], in the context of MRI denoising, which is different from MRI acceleration problem. The works in [43, 66, 111] adopt a Tucker model [143], and also require auxiliary acquisition data to estimate the bases in non-spatial modes prior to reconstruction, which are not available in most fMRI acquisitions.

Finally, the work in [16], adopts a random sampling t-SVD algorithm to handle a single coil MRI acceleration process, which is not applicable to multi-coil acquisitions used in practice.

4.6 General Algorithmic framework for Tensor Sampling

Previously we studied the identifiability of third-order tensors under different sampling mechanisms. In the current section, the algorithmic component of our approach is discussed. In general, reconstruction of a tensor from a subset of entries falls under the framework of tensor completion. A plethora of algorithms have been proposed, e.g. [3, 158]. The idea is to use the CPD factors, computed from the incomplete tensor, and reconstruct the original one. Popular methods approach the problem as a system of non-linear equations and handle it using descent direction approaches, such as gradient descent, alternating optimization, or the Gauss-Newton method.

Existing tensor completion works could be employed to approach the recovery task of a regularly sampled tensor. However, the unique characteristics of our models would be ignored. To put it in context, the special structure of *regular sampling allows tensor completion by computing the factors of complete tensors*. Note that CPD computation of a complete tensor is a considerably easier task than that of an incomplete one. Several polynomial time algebraic algorithms [48, 136] have been shown to retrieve the original factors, under certain conditions, or effectively initialize optimization approaches with significant success.

We propose a three step approach to tackle the completion task, which follows the insights of Theorems 4.1-4.5. The first step solves the CPD of the sub-sampled tensors independently, the second reconciles for permutation/scaling ambiguities and gets an initial estimate of the factors, and the third solves the coupled CPD problem. Detailed analysis follows.

4.6.1 Step 1: Computing the CPD of sub-tensors

First, the CPD of the sub-sampled tensors $\underline{\boldsymbol{Y}}_i$ is computed. This step is guided from the requirements of each sampling mechanism. To be more precise, CPD of $\underline{\boldsymbol{Y}}_i$ is computed if the reconstruction conditions require $\underline{\boldsymbol{Y}}_i$ to admit an essentially unique CPD. The slab sampling model, for instance, requires only $\underline{\boldsymbol{Y}}_1$ or $\underline{\boldsymbol{Y}}_2$ to admit unique CPD. Therefore the CPD of only one sub-tensor is needed. On the other hand when fiber or entry sampling is considered, the CPD computation of every sub-tensor is performed, following the previous identifiability analysis.

4.6.2 Step 2: Initializing the factors

After computing the CPD of the sub-tensors, step 2 computes an initial estimate of the $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ factors, after resolving possible permutation and scaling mismatches. We distinguish between 2 different cases:

Case 1, slab sampling: As mentioned earlier slab sampling automatically reconciles for permutation and scaling ambiguities. Furthermore, two of the factors have been already computed from step 1 (e.g., $\boldsymbol{B}, \boldsymbol{C} \leftarrow \text{CPD}(\underline{\boldsymbol{Y}}_1)$). What remains to be obtained is the third factor (e.g., $\boldsymbol{A}$), which is revealed by the other sub-tensor ($\underline{\boldsymbol{Y}}_2$), via solving a linear system of equations:

$$\boldsymbol{Y}_2^{(1)} = (\boldsymbol{P}_3^{(2)}\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^T.$$

Case 2, fiber and entry sampling: Contrary to slab sampling, the permutation and scaling ambiguity is an important issue when fiber or entry sampling is applied. To be more precise, let $\underline{\boldsymbol{Y}}_d = [\![\boldsymbol{A}_d, \boldsymbol{B}_d, \boldsymbol{C}_d]\!],\ d \in \{1, \dots D\}$, be the sub-tensors formed after fiber or entry sampling. Then:

$$\boldsymbol{A}_d = \boldsymbol{P}_1^{(d)}\boldsymbol{A}\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_1^{(d)} = \boldsymbol{A}(\mathcal{S}_r^{(d)}, :)\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_1^{(d)}, \tag{4.9a}$$

$$\boldsymbol{B}_d = \boldsymbol{P}_2^{(d)}\boldsymbol{B}\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_2^{(d)} = \boldsymbol{B}(\mathcal{S}_c^{(d)}, :)\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_2^{(d)}, \tag{4.9b}$$

$$\boldsymbol{C}_d = \boldsymbol{P}_3^{(d)}\boldsymbol{C}\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_3^{(d)} = \boldsymbol{C}(\mathcal{S}_f^{(d)}, :)\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_3^{(d)}, \tag{4.9c}$$

where $\boldsymbol{\Pi}^{(d)} \neq \boldsymbol{\Pi}^{(d')}$ are permutation matrices and $\boldsymbol{\Lambda}_i^{(d)} \neq \boldsymbol{\Lambda}_i^{(d')}$ are full rank diagonal matrices such that $\boldsymbol{\Lambda}_1^{(d)}\boldsymbol{\Lambda}_2^{(d)}\boldsymbol{\Lambda}_3^{(d)} = \boldsymbol{I},\ d, d' \in \{1, \dots, D\},\ d' \neq d$. Clearly, in order to synthesize $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ from $\boldsymbol{A}_d, \boldsymbol{B}_d, \boldsymbol{C}_d$ and reconstruct $\underline{\boldsymbol{X}}$, the permutation and scaling mismatch should be resolved, i.e., $\boldsymbol{\Pi}^{(d)} = \boldsymbol{\Pi}^{(d')}$, $\boldsymbol{\Lambda}_i^{(d)} = \boldsymbol{\Lambda}_i^{(d')}$ for every $d,\ d'$.

To overcome this issue, the common information between sub-tensors is utilized. In simple words, (4.4) or (4.6) require $\boldsymbol{C}_d - \boldsymbol{C}_{d'}$ (or $\boldsymbol{A}_d - \boldsymbol{A}_{d'}$, or $\boldsymbol{B}_d - \boldsymbol{B}_{d'}$) to share some common rows, i.e., $|\mathcal{S}_f^{(d)} \cap \mathcal{S}_f^{(d')} := \mathcal{S}_f^{(d-d')}| \geq 2$. Now let:

$$\boldsymbol{C}_d^{(d-d')} = \boldsymbol{C}(\mathcal{S}_f^{(d-d')}, :)\boldsymbol{\Pi}^{(d)}\boldsymbol{\Lambda}_3^{(d)} \tag{4.10a}$$

$$\boldsymbol{C}_{d'}^{(d-d')} = \boldsymbol{C}(\mathcal{S}_f^{(d-d')}, :)\boldsymbol{\Pi}^{(d')}\boldsymbol{\Lambda}_3^{(d')} \tag{4.10b}$$

and normalize $\boldsymbol{C}_d^{(d-d')},\ \boldsymbol{C}_{d'}^{(d-d')}$, such that they share a common row with same scaling, but permutation mismatch, e.g.:

$$\bar{\boldsymbol{C}}_d^{(d-d')} = \boldsymbol{C}_d^{(d-d')}\boldsymbol{G}_d^{-1},\ \bar{\boldsymbol{C}}_{d'}^{(d-d')} = \boldsymbol{C}_{d'}^{(d-d')}\boldsymbol{G}_{d'}^{-1},$$

where $\boldsymbol{G}_d = \text{diag}\left(\boldsymbol{C}_d^{(d-d')}(1,:)\right)$, $\boldsymbol{G}_{d'} = \text{diag}\left(\boldsymbol{C}_{d'}^{(d-d')}(1,:)\right)$ and $\text{diag}(\mathbf{x})$ is the diagonal matrix of row vector $\mathbf{x}$. Then:

$$\bar{\boldsymbol{C}}_d^{(d-d')} = \bar{\boldsymbol{C}}_{d'}^{(d-d')}\boldsymbol{\Pi}^{(d')^{-1}}\boldsymbol{\Pi}^{(d)} = \bar{\boldsymbol{C}}_{d'}^{(d-d')}\bar{\boldsymbol{\Pi}} \tag{4.11}$$

and we can solve for $\bar{\boldsymbol{\Pi}}$ using the Hungarian algorithm [99]. This procedure resolves the permutation mismatch between the factors, i.e., $\boldsymbol{\Pi}^{(d)} = \boldsymbol{\Pi}^{(d')}$. Note that in case of fiber sampling $\mathcal{S}_f^{(d)} = \mathcal{S}_f^{(d')} = \mathcal{S}_f^{(d-d')} = \{1 \dots K\}$.

To reconcile for the scaling ambiguity we require extra information coming from the factors that were not involved in permutation match, i.e., $\boldsymbol{A}_d - \boldsymbol{A}_{d'}$ or $\boldsymbol{B} - \boldsymbol{B}_{d'}$ in our example. The necessary rules (4.4c), (4.6d) enforce that there is at least one common row between $\boldsymbol{A}_d - \boldsymbol{A}_{d'}$ or $\boldsymbol{B} - \boldsymbol{B}_{d'}$. Then the scaling mismatch can be solved by the following set of equations:

$$\boldsymbol{A}_d^{(d-d')} = \boldsymbol{A}_{d'}^{(d-d')}\boldsymbol{\Lambda}_1^{(d')^{-1}}\boldsymbol{\Lambda}_1^{(d)} \tag{4.12a}$$

$$\boldsymbol{B}_d^{(d-d')} = \boldsymbol{B}_{d'}^{(d-d')}\boldsymbol{\Lambda}_2^{(d')^{-1}}\boldsymbol{\Lambda}_2^{(d)} \tag{4.12b}$$

$$\boldsymbol{C}_d^{(d-d')} = \boldsymbol{C}_{d'}^{(d-d')}\boldsymbol{\Lambda}_3^{(d')^{-1}}\boldsymbol{\Lambda}_3^{(d)} \tag{4.12c}$$

$$\boldsymbol{\Lambda}_1^{(d)}\boldsymbol{\Lambda}_2^{(d)}\boldsymbol{\Lambda}_3^{(d)} = \boldsymbol{I},\ \boldsymbol{\Lambda}_1^{(d')}\boldsymbol{\Lambda}_2^{(d')}\boldsymbol{\Lambda}_3^{(d')} = \boldsymbol{I} \tag{4.12d}$$

$\boldsymbol{A}_d^{(d-d')},\ \boldsymbol{A}_{d'}^{(d-d')}$ and $\boldsymbol{B}_d^{(d-d')},\ \boldsymbol{B}_{d'}^{(d-d')}$ represent the common rows between $\boldsymbol{A}_d - \boldsymbol{A}_{d'}$ and $\boldsymbol{B} - \boldsymbol{B}_{d'}$ respectively. Next, an initial estimate of the factors is extracted by reading out the appropriate rows from the sub-tensor factors to synthesize $\boldsymbol{A},\ \boldsymbol{B},\ \boldsymbol{C}$ i.e.,

$$\boldsymbol{A}(\mathcal{S}_r^{(d)},:) \leftarrow \boldsymbol{A}_d,\ \boldsymbol{B}(\mathcal{S}_c^{(d)},:) \leftarrow \boldsymbol{B}_d,\ \boldsymbol{C}(\mathcal{S}_f^{(d)},:) \leftarrow \boldsymbol{C}_d,\ \forall d.$$

4.6.3 Step 3: Coupled CPD

Finally, $\boldsymbol{A},\ \boldsymbol{B},\ \boldsymbol{C}$ are jointly computed as a classic tensor factorization problem with missing entries, which is equivalent to the following coupled CPD estimator:

$$\text{minimize}_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \sum_{d=1}^{D} \left\| \underline{\boldsymbol{Y}}_d - \left[\!\left[\boldsymbol{P}_1^{(d)}\boldsymbol{A}, \boldsymbol{P}_2^{(d)}\boldsymbol{B}, \boldsymbol{P}_3^{(d)}\boldsymbol{C} \right]\!\right] \right\|_F^2 . \tag{4.13}$$

In slab sampling, $\boldsymbol{P}_2^{(1)}, \boldsymbol{P}_3^{(1)}, \boldsymbol{P}_1^{(2)}, \boldsymbol{P}_2^{(2)}$ are identity matrices and in fiber sampling $\boldsymbol{P}_3^{(d)}$ is always the identity. There are several ways to handle the above non-convex problem. We choose to employ the tensorlab [166] toolbox, which uses a Gauss Newton approach to solve this nonlinear least squares (NLS) problem. After obtaining the estimates of $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$, $\underline{\boldsymbol{X}}$ can be reconstructed by:
$\underline{\hat{\boldsymbol{X}}}(i,j,k) = \sum_{f=1}^{F} \hat{\boldsymbol{A}}(i,f)\hat{\boldsymbol{B}}(j,f)\hat{\boldsymbol{C}}(k,f)$.

4.6.4 REgular Tensor Sampling and INterpolation Algorithm (RETSINA)

As mentioned earlier the accelerated fMRI acquisition can be cast as a tensor sampling and reconstruction task. Therefore it falls under the class of problems that the previously described framework can handle. However we choose to follow a different initialization approach tailored to the specific application. We design two algorithms, one for single-slice fMRI and one for multi-slice fMRI. In both algorithms, $\underline{\boldsymbol{W}}$ denotes the sampling mask, i.e., $\boldsymbol{W}(i,j,k) = 1$, if $\underline{\boldsymbol{X}}(i,j,k)$ is sampled / observed and $\boldsymbol{W}(i,j,k) = 0$ otherwise. $\underline{\tilde{\boldsymbol{X}}}$ is the incomplete tensor from which we form $\underline{\boldsymbol{Y}}_d$ for $d = 1, \ldots D$ and $*$ denotes the Hadamard product.

Single slice acceleration: The REgular Tensor Sampling and INterpolation Algorithm (`RETSINA`) is presented in Algorithm 4.1. We follow a 3 step procedure. In step 1 (initialization), for n-fold acceleration we sum every n vertical slabs (frames), where the missing k-space measurements are considered zeros, and obtain a tensor $\underline{\boldsymbol{X}}_n \in \mathbb{C}^{I \times J/n \times K}$ without missing entries. Then we compute the CPD of $\underline{\boldsymbol{X}}_n$ to get a rough estimate of $\boldsymbol{A},\ \boldsymbol{C}$ factors and solve d linear systems of equations to approximate $\boldsymbol{B}$. In step 2 (refinement), we compute the CPD of $\underline{\boldsymbol{Y}}_1$ (initialized by step 1) and the CPD of $\{\underline{\boldsymbol{Y}}_d\}_{d\neq 1}$ with known $\boldsymbol{C}$. The number of iterations in step 2 should remain low (e.g., 2), to maintain the permutation and scaling matching between the factors of sub-tensors $\{\underline{\boldsymbol{Y}}_d\}$. Finally, in step 3, we compute the final factors by solving (4.13) with tensorlab's Gauss-Newton algorithm. Compared to the previously presented general

framework, RETSINA empirically yields enhanced reconstruction accuracy and reduces the operational time.

Algorithm 4.1 RETSINA

Input: $n,\ F,\ \underline{\tilde{\boldsymbol{X}}},\ \underline{\boldsymbol{W}}$.

step 1: Initialization;

$\underline{\boldsymbol{X}}_n(:,j,:) = \sum_{l=(j-1)n+2}^{jn+1} (\underline{\boldsymbol{W}} * \underline{\tilde{\boldsymbol{X}}})(:,l,:)$.

$\boldsymbol{A}, \boldsymbol{C} \leftarrow \text{CPD}(\underline{\boldsymbol{X}}_n)$.

Form $\{\underline{\boldsymbol{Y}}_d, \mathcal{S}_r^{(d)}, \mathcal{S}_c^{(d)}\}_{d=1}^{D}$ from $\underline{\tilde{\boldsymbol{X}}}$.

$\boldsymbol{B}(\mathcal{S}_c^{(d)},:) = \arg\min_{\boldsymbol{Z}} \|\underline{\boldsymbol{Y}}_i - [\![\boldsymbol{A}(\mathcal{S}_r^{(d)},:), \boldsymbol{Z}, \boldsymbol{C}]\!]\|_F^2$.

step 2: Refinement;

$\boldsymbol{A}(\mathcal{S}_r^{(1)},:), \boldsymbol{B}(\mathcal{S}_c^{(1)},:), \boldsymbol{C} \leftarrow \text{CPD}(\underline{\boldsymbol{Y}}_1)$.

$\boldsymbol{A}(\mathcal{S}_r^{(d)},:), \boldsymbol{B}(\mathcal{S}_c^{(d)},:), \sim\leftarrow \text{CPD}(\underline{\boldsymbol{Y}}_d), \quad d \neq 1$.

step 3: Solve (4.13) using Gauss-Newton.

Reconstruct the missing entries of $\underline{\boldsymbol{X}}$ using $\underline{\hat{\boldsymbol{X}}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$.

Algorithm 4.2 MS-RETSINA

Input: $\rho,\ s,\ F,\ \underline{\tilde{\boldsymbol{X}}},\ \underline{\boldsymbol{W}}$.

step 1: Initialization;

$\underline{\boldsymbol{X}}_n(:,j,:) = \sum_{l=(j-1)n+2}^{jn+1} (\underline{\boldsymbol{W}} * \underline{\tilde{\boldsymbol{X}}})(:,l,:), n = \rho s$.

$\boldsymbol{A}, \boldsymbol{C} \leftarrow \text{CPD}(\underline{\boldsymbol{X}}_n)$.

Form $\{\underline{\boldsymbol{Y}}_d, \mathcal{S}_r^{(d)}, \mathcal{S}_c^{(d)}, \mathcal{S}_f^{(d)}\}_{d=1}^{D}$ from $\underline{\tilde{\boldsymbol{X}}}$.

$\boldsymbol{B}(\mathcal{S}_c^{(d)},:) = \arg\min_{\boldsymbol{Z}} \|\underline{\boldsymbol{Y}}_i - [\![\boldsymbol{A}(\mathcal{S}_r^{(d)},:), \boldsymbol{Z}, \boldsymbol{C}]\!]\|_F^2$.

step 2: Solve (4.13) using Gauss-Newton.

Reconstruct the missing entries of $\underline{\boldsymbol{X}}$ using $\underline{\hat{\boldsymbol{X}}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$.

Multi slice acceleration: The Multi-Slice RETSINA (MS-RETSINA) is presented in Algorithm 4.2. Compared to RETSINA the initialization step has been modified to this specific case and the refinement step is skipped, since it has been observed that it does not improve the overall performance.

4.7 Simulations

In this section, we showcase the effectiveness of the proposed tensor sampling framework using numerical experiments. The experiments involve synthetically generated data as well as fMRI scans in the k-space. All simulations are performed in MATLAB on a Linux server with 3.6GHz cores and 32GB RAM, except part C which is performed on a Linux server with 2GHz cores and 128GB RAM. All CPD computations required in our proposed algorithms are performed using Tensorlab's non-linear least squares algorithm, which combines algebraic initialization and Gauss-Newton iterations.

4.7.1 Synthetic Experiments

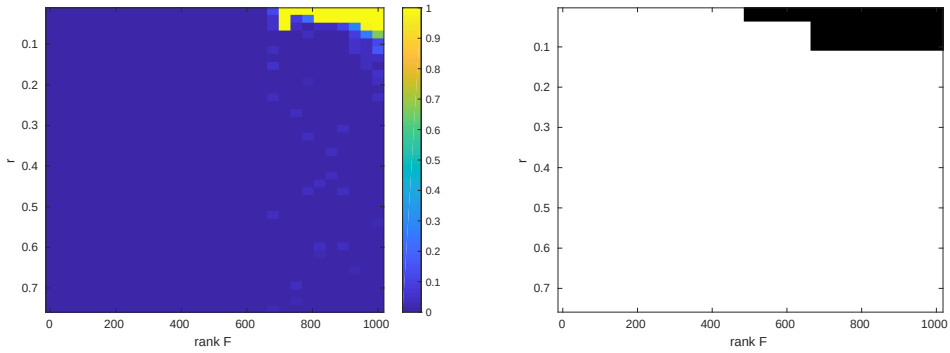
First synthetically generated experiments are conducted to examine the validity of our claims and the performance of the proposed framework. In particular, a tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{512\times512\times512}$ is generated as $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. The elements of the factor matrices are drawn from an independent identically distributed (i.i.d.) zero mean, unit variance Gaussian distribution. We regularly sample $\underline{\boldsymbol{X}}$ according to the previously presented sampling mechanisms, i.e,. slab sampling, fiber sampling, and entry sampling, as shown in Figs. 4.1- 4.4. Specifically, for slab sampling we sample equispaced frontal and horizontal slabs. Regarding fiber sampling, each tensor $\underline{\boldsymbol{Y}}_i$ is a set of fibers, defined by equispaced rows and columns of $\underline{\boldsymbol{X}}$. Note that one vertical slab is fully observed to reconcile for permutation and scaling ambiguities. Equivalently entry sampling is designed to observe different sets of equispaced entries plus a fully sampled vertical slab.

For the experiments, we vary the sampling ratio, i.e., $r = \frac{\#\text{of sampled entries}}{IJK}$, from 0.75 to 0.001. We also vary the tensor rank F from 5 to 1000. To evaluate the performance of tensor reconstruction, we measure the normalized reconstruction error, i.e.

$$\texttt{NRE} = \frac{\sum_{k=1}^{K} \|\underline{\hat{\boldsymbol{X}}}(:,:,k) - \underline{\boldsymbol{X}}(:,:,k)\|_F}{\sum_{k=1}^{K} \|\underline{\boldsymbol{X}}(:,:,k)\|_F}$$

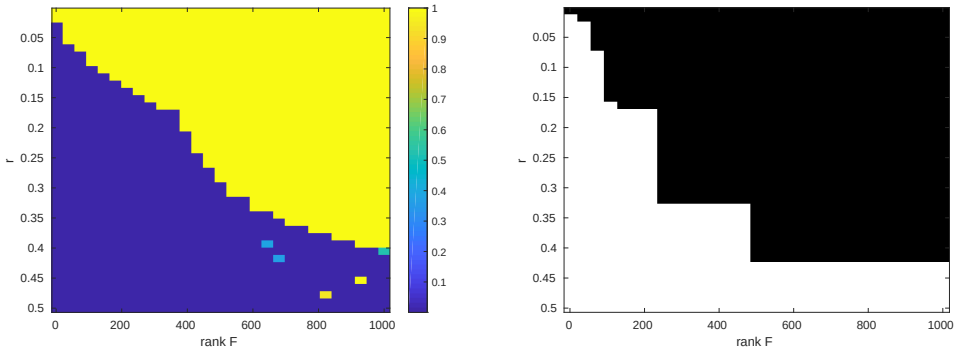
When $\texttt{NRE} > 1$ we set $\texttt{NRE} = 1$, so that our 2-dimensional plot clearly shows the regions where completion is successful and regions where completion fails. Figs. 4.7-4.9 present the results for the three sampling schemes. The left column of each figure illustrates the NRE of reconstruction. The right column shows the identifiability threshold, for each experiment, derived

by Theorem 4.1, 4.2 or 4.3, according to the applied sampling mechanism. Identifiability is guaranteed almost surely in white regions and in black regions our sufficient conditions are not satisfied. As expected the reconstruction accuracy is deteriorating as the rank F increases or



(a) NRE (b) Identifiability condition

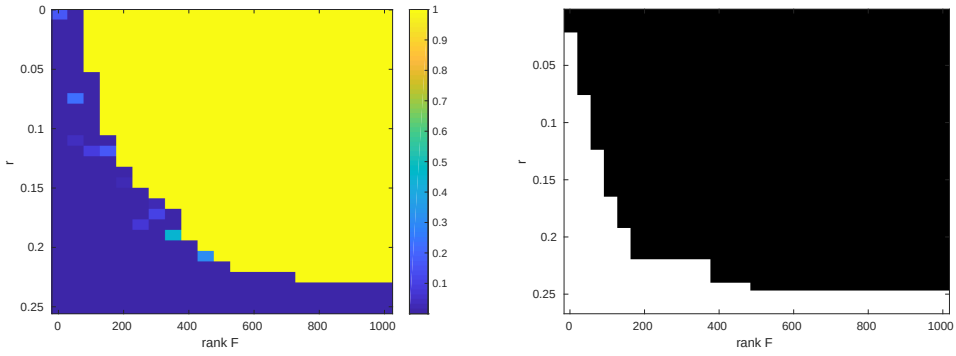
Figure 4.7: rank F vs sampling ratio r for slab sampling.



(a) NRE (b) Identifiability condition

Figure 4.8: rank F vs sampling ratio r for fiber sampling.

the sampling ratio r decreases. For reasonably small ranks and high number of samples the reconstruction is perfect. As far as identifiability is concerned we observe that for slab and entry sampling the identifiability threshold follows an analogous trend to that of NRE, whereas for fiber sampling the transitions in the identifiability trend are not as smooth as the transitions in the reconstruction trend. Furthermore, in the vast majority of considered cases, reconstruction is successful when the identifiability conditions are satisfied. There exist cases, however, where the



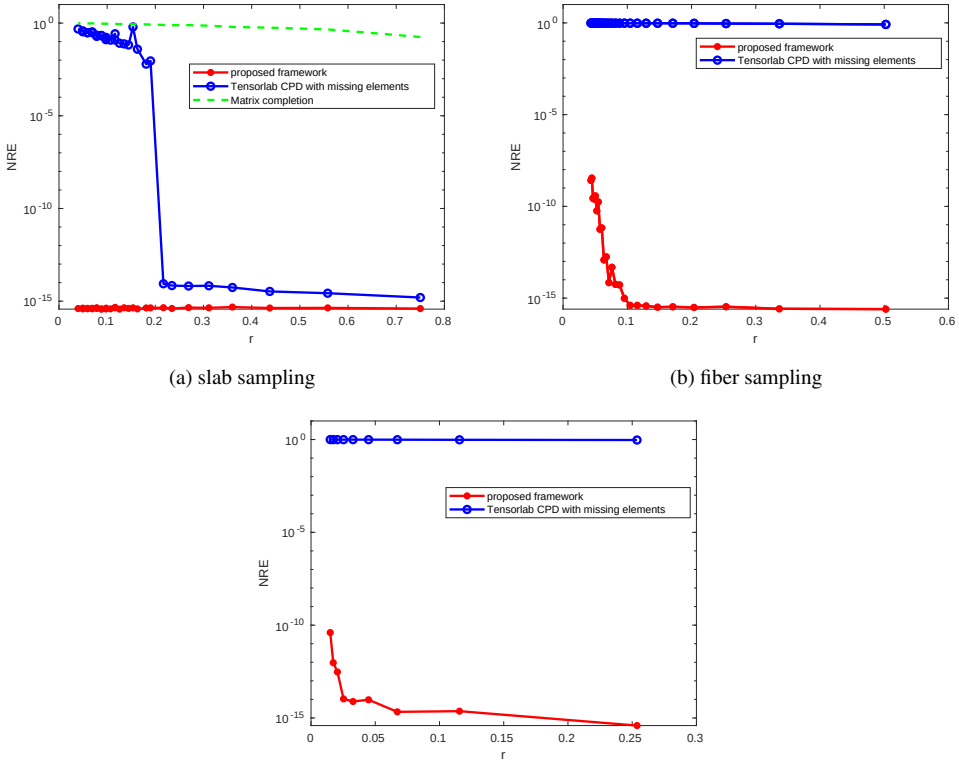
(a) NRE (b) Identifiability condition

Figure 4.9: rank F vs sampling ratio r for entry sampling.

proposed identifiability conditions are not satisfied, but reconstruction is successful, especially when fiber or entry sampling is applied. This is expected, since our conditions are sufficient and not necessary. We also observe that it appears to be a sudden transition between $\texttt{NRE} \approx 0$ and $\texttt{NRE} \approx 1$. This happens due to the fact that the proposed framework initially solves the CPD of the subtensors $\underline{\boldsymbol{Y}}_d$. When the combination of rank and number of samples is close to the identifiability threshold, we observed that some of the subtensors $\underline{\boldsymbol{Y}}_d$ yield CPD solutions that are not unique and thus permutation and scaling matching fails, which leads to bad estimates of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and high values of NRE.

Next, we compare the reconstruction performance of our proposed framework with a classic tensor completion approach for general sampling patterns, and a matrix completion approach that operates on each vertical slab separately. To this end, we generate a tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{200\times200\times200}$ with rank $F = 20$, where the elements of factors $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are drawn independently from a zero mean, unit variance Gaussian distribution. We apply the previously described sampling mechanisms for different levels of downsampling. The algorithm employed for general tensor completion is Tensorlab's CPD algorithm with missing elements. For matrix completion of each slab we use a nuclear norm minimization algorithm (referred to as Matrix completion) implemented in TFOCS [21], which is a powerful and flexible first order framework for convex optimization problems. Note that when regular fiber and entry sampling is applied to a third order tensor, entire columns or rows of each slab are likely to remain unobserved; see Figs. 3-6 to better appreciate this point. As a result any nuclear norm minimization or matrix factorization based approach is guaranteed to fail. Therefore we limit our comparison with matrix completion only

to the case of slab sampling. Fig. 4.10 illustrates the performance of the competing algorithms. From Fig. 4.10, it is clear that although the considered scenarios are effectively handled by our



(a) slab sampling

(b) fiber sampling

(c) entry sampling

Figure 4.10: Completing a tensor with different methods.

proposed framework, classic methods fail in producing satisfactory results in the majority of the cases. Tensor completion via Tensorlab's CPD with missing elements is only successful for high sampling ratios of slab sampling, while both matrix and Tensorlab completion fail in the rest.

4.7.2 Accelerated parallel fMRI

Next, the tensor sampling and reconstruction framework is tested in a real and important problem, that of parallel fMRI acceleration. First, we test the performance of the proposed `RETSINA` with fMRI scans, fully sampled in the k-space, obtained from the Center for Magnetic Resonance Research (CMRR) at the University of Minnesota.

The single slice raw scan is originally a fourth-order tensor of size $104 \times 104 \times 32 \times 490$ and we unfold it as a third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{C}^{10816 \times 32 \times 490}$. We apply 3-fold acceleration by observing $1/3$ of the k_y frequencies, as shown in Fig. 4.5. We choose $F = 100$ and run step 1, 2, 3 of Algorithm 1 for 50, 2 and 5 iterations respectively. The baseline algorithms used for comparison are `k-t Focuss` [79], which is a CS type algorithm, `k-t SLR` [106] which combines ideas from both LRMC and CS, and the zero padding inverse discrete Fourier transform (IDFT). We also compare with general tensor completion algorithms, i.e., Tensorlab's CPD with missing elements (`T-CPD` with miss.), Tensorlab's Tucker with missing elements (`T-Tucker` with miss.) and `t-SVD` [190]. For `T-CPD`, we choose $F = 100$ and `T-Tucker` the core dimensions are set to $(50, 50, 10)$. The maximum number of iterations for the three general-purpose tensor completion algorithms is set to 100. Note that the performance of IDFT is an indicator on how difficult the reconstruction is. It is also worth noticing that `k-t SLR` directly reconstructs the fMRI signal in the absolute $(x-y)$-time-coil space. To be more precise it reconstructs signal $\underline{\boldsymbol{H}} = |\mathcal{Q}(\underline{\boldsymbol{X}})|$, where $\mathcal{Q}$ denotes the inverse Fourier transform from the $k_x - k_y$ to the $x - y$ space and $|\cdot|$ is the absolute value. Thus we also measure the NRE of signal $\underline{\boldsymbol{H}}$, denoted as NRE_2, for fair comparisons. For `k-t Focuss`, `k-t SLR` and `t-SVD` the publicly available code was used. Note that `k-t Focuss` and `k-t SLR` are single coil algorithms in their original implementation, thus we treated each coil separately. `k-t SLR` requires parameter tuning and so we used a validation step to tune effectively.

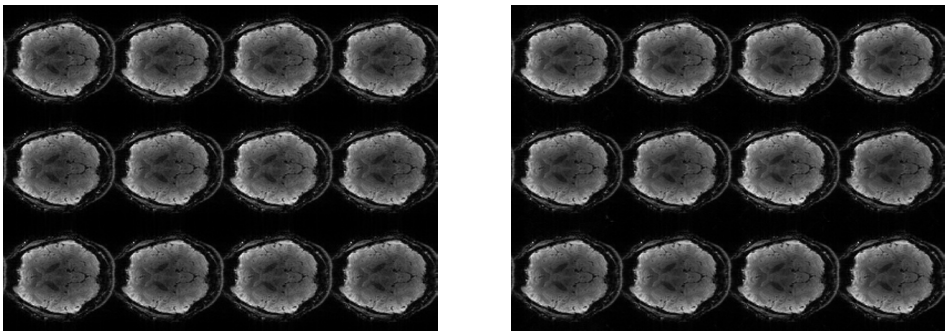
The results are presented in Table 4.1, which includes the NRE in k-space and absolute $x-y$ space as well as runtime. The proposed `RETSINA` achieves highest reconstruction quality in the k-space and works comparably well (but markedly faster) with `k-t SLR` in reconstructing the signal magnitude in the $x - y$ space. This is expected, since `RETSINA` reconstructs both the magnitude and phase (which is very important in images) in the k-space, whereas `k-t SLR` reconstructs the magnitude in the $x - y$ space. In terms of runtime `RETSINA` works faster than `k-t SLR` and `k-t Focuss`, but slower than `IDFT`. However `IDFT` exhibits very poor reconstruction performance. It is worth noting, that `k-t Focuss` and `k-t SLR` are amenable to parallel implementation, which could speed up their computation at the cost of additional hardware. Regarding the general-purpose tensor completion algorithms, it is clear that all of them fail to produce satisfactory reconstruction results and also require significant computation time. This result highlights the challenging nature of regular sampling and the need for a customized framework to tackle it. Note that for all algorithms we used a 32GB RAM server to perform

these experiments, except t−SVD which exhausted all the memory resources and required the use of the 128 GB RAM server.

Table 4.1: Reconstruction performance of the competing algorithms.

Algorithm	NRE	NRE_2	runtime
RETSINA	**0.124**	0.081	8min
k−t Focuss	0.339	0.286	25.6min (48sec/coil)
k−t SLR	1.41	**0.073**	480min (15min/coil)
IDFT	0.8156	0.7376	**14sec**
T−CPD with miss.	0.7570	0.6812	118min
T−Tucker with miss.	0.7150	0.6397	65min
t−SVD	0.7630	0.6818	627min

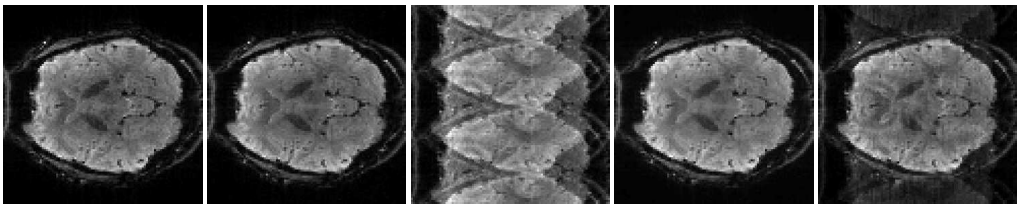
Fig. 4.11 shows the reconstructed fMRI scans at different time frames produced by RETSINA along with the fully sampled data. The quality of the reconstruction is significantly high, rendering the proposed RETSINA a good alternative for fMRI acceleration. Finally, in Fig. 4.12



(a) fully sampled scan (b) RETSINA

Figure 4.11: fMRI reconstruction with 3-fold acceleration

we illustrate the reconstruction performance at a single frame for the competing algorithms. IDFT gives an illustration of the downsampled image, RETSINA works the best and k−t SLR work comparably well, although being slightly off in contrast.



(a) original (b) RETSINA (c) IDFT (d) k-t SLR (e) k-t Focuss

Figure 4.12: Reconstruction at a single frame

4.7.3 Accelerated multi-slice parallel fMRI

Finally, the proposed framework is tested in the task of accelarated multi-slice fMRI acquisition. Recall that acceleration is performed at 2 levels, since at each time slot we measure the sampled k-space of only a subset of slices. The multi-slice fMRI raw scan is a fifth-order tensor of size $104 \times 104 \times 32 \times 490 \times 8$, where the number of slices is 8. We unfold it as a third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{C}^{10816 \times 490 \times 256}$ and observe $1/\rho$ of the k_y frequencies and $1/s$ of slices, which leads to ρs-fold acceleration, as illustrated in Fig. 4.6. The tensor rank used in MS−RETSINA is $F = 100$ and the maximum number of iterations in step 1 and step 2 are set to 50 and 20 respectively. Table 4.2 shows the performance of the proposed MS−RETSINA in terms of NRE for various values of ρ and s. An illustrative example of the reconstruction performance when $\rho = s = 2$ is presented in Fig. 4.13.

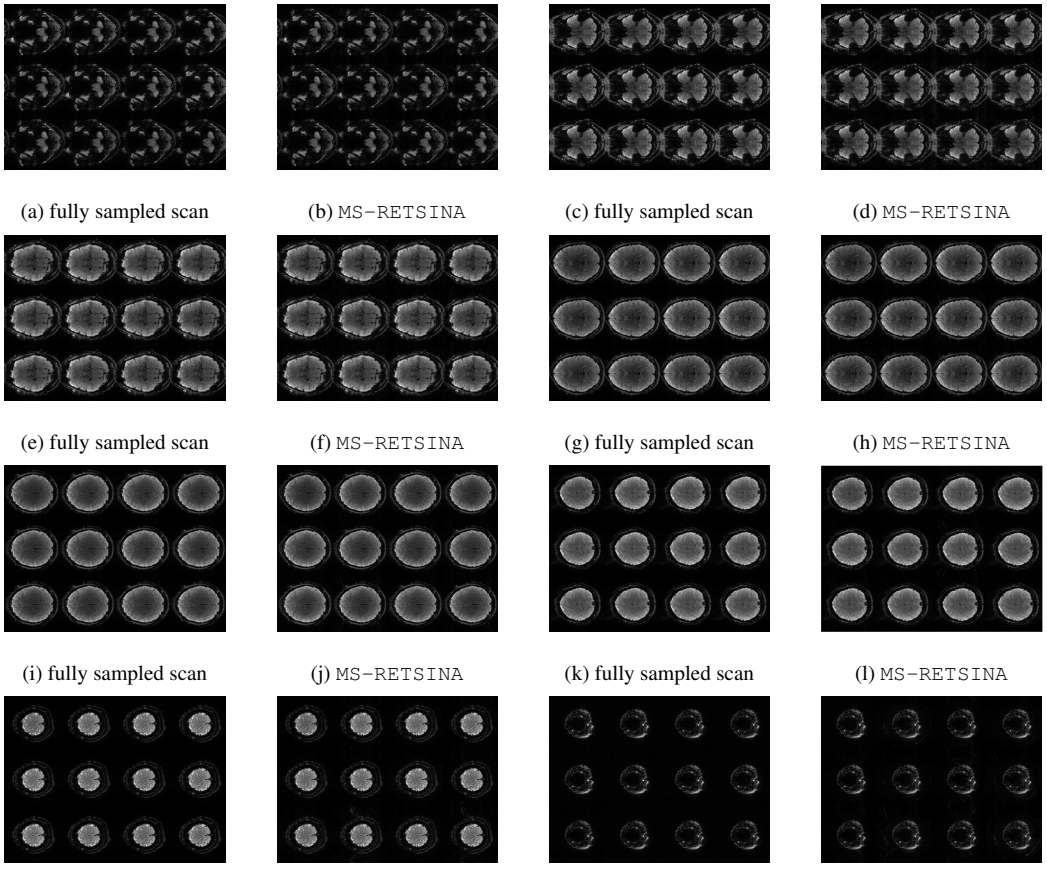
Table 4.2: NRE performance of MS−RETSINA.

$s\backslash\rho$	2	3	4	5	6	7	8	9	10
2	**0.14**	0.15	0.17	0.19	0.18	0.18	0.18	0.18	0.20
3	0.17	0.18	0.19	0.19	0.19	0.21	0.20	0.19	0.21
4	0.19	0.20	0.20	0.22	0.21	0.21	0.23	0.21	0.23

4.8 Conclusion

In this chapter we studied the sampling and reconstruction of tensors under various schemes. Compared to CS, LRMC, as well as other tensor works, we provide concrete conditions, deterministic and generic, under which tensor completion from regular samples is identifiable.

Furthermore, we cast the fMRI acceleration task as regular tensor sampling process and provided an efficient algorithmic framework to approach the problem. Simulations with synthetic data as well as fMRI scans in the k-space show the validity and effectiveness of our approach.



(a) fully sampled scan (b) MS−RETSINA (c) fully sampled scan (d) MS−RETSINA

(e) fully sampled scan (f) MS−RETSINA (g) fully sampled scan (h) MS−RETSINA

(i) fully sampled scan (j) MS−RETSINA (k) fully sampled scan (l) MS−RETSINA

(m) fully sampled scan (n) MS−RETSINA (o) fully sampled scan (p) MS−RETSINA

Figure 4.13: fMRI reconstruction with 4-fold acceleration

Chapter 5

Large-scale Canonical Polyadic Decomposition via Regular Tensor Sampling

Large-scale, multidimensional tensors are ubiquitous in various engineering domains. Computing the canonical polyadic decomposition (CPD) of these tensors is an important task, since CPD is an effective analysis tool in various applications, including signal processing, machine learning, and communications, to name a few. However, when the tensor size gets big, computing the CPD becomes a lot more challenging.

Previous works proposed using random (generalized) tensor sampling or compression to alleviate this challenge. Albeit the number and variety of works in computing the decomposition of large-scale tensors, there are still remaining challenges that need to be addressed. First, it is often the case that model identifiability is not discussed, especially in works that use sampling to facilitate the computation. Note that model identifiability is important, since it guarantees that the solution of the computationally lighter problem is the same as the solution of the original one in the noiseless case (or, fixing residuals). Furthermore, although there exist various effective algorithms for big sparse tensors, this is not the case for big and dense ones, which leaves room for additional improvement. Finally, a number of existing works exhibit significant performance drop, when real, noisy data are involved and thus there is need for alternative approaches.

In this chapter, we propose using a regular tensor sampling framework instead. We show

that by appropriately selecting the sampling mechanism, we can simultaneously control memory and computational complexity, while guaranteeing identifiability at the same time. Numerical experiments with synthetic and real data showcase the effectiveness of our approach. Part of the work presented in this Chapter is published in [86].

5.1 Prior Art

Various works have been proposed to efficiently compute the CPD of big data tensors. A first class of algorithms focused in efficiently computing the CPD of big sparse tensors [14, 89, 124, 149]. The work in [124], for example, uses a random sampling mechanism to compute the non-negative CPD of sparse tensors, whereas [149] uses a novel, memory friendly sparse tensor data structure in conjunction with parallel implementation. The idea of random sampling has also been used in computing more general tensor structures. The work in [167] uses randomized block sampling to update only a subset of affected variables at each iteration, thus mitigating the computational burden. Tensor compression is another idea which has been used instead of sampling. First compression was applied via the higher-order singular value decomposition (HOSVD) [45], followed by [145, 182], which create compressed versions of the big tensor by multiplying it with random and pseudo-random matrices respectively. Finally the idea of computing the CPD of an incomplete version of the big tensor has been considered in [169].

5.2 Sampling in multiple modes

We begin our discussion by presenting the sampling schemes, used to compute the CPD of large-scale tensors. We propose two sampling mechanisms, which operate on multiple modes of the tensor, and provide identifiability analysis of the sampling model.

5.2.1 Combining slab and fiber sampling

The first sampling mechanism, for CPD computation purposes, combines slab and fiber sampling. In particular, we propose to subsample a subset of frontal slabs $\mathcal{S}_f \subseteq \{1, \dots, K\}$, along with a subset of fibers, defined by rows $\mathcal{S}_r \subseteq \{1, \dots, I\}$ and columns $\mathcal{S}_c \subseteq \{1, \dots, J\}$. Then two

sub-sampled tensors are formed:

$$\underline{\boldsymbol{Y}}_1 = \underline{\boldsymbol{X}}\left(\mathcal{S}_r, \mathcal{S}_c, :\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 \tag{5.1}$$

$$\underline{\boldsymbol{Y}}_2 = \underline{\boldsymbol{X}}\left(:, :, \mathcal{S}_f\right) = \underline{\boldsymbol{X}} \times_3 \boldsymbol{P}_3, \tag{5.2}$$

where $\boldsymbol{P}_1 \in \mathbb{R}^{I_1 \times I}$, $\boldsymbol{P}_2 \in \mathbb{R}^{J_1 \times J}$, $\boldsymbol{P}_3 \in \mathbb{R}^{K_2 \times K}$ are row, column and fiber selection matrices corresponding to $\mathcal{S}_r$, $\mathcal{S}_c$, $\mathcal{S}_f$ respectively. An illustration of this sampling technique is depicted in Fig. 5.1. Note that the samples are not drawn arbitrarily. In contrast with [124, 167] the

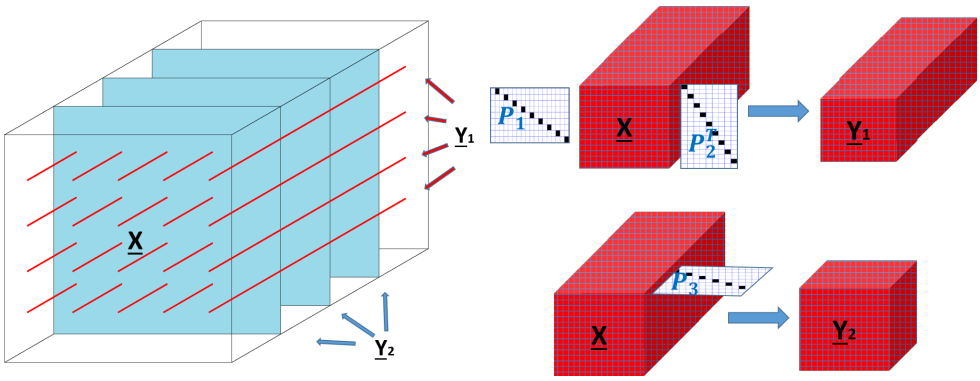


Figure 5.1: Combination of fiber and frontal slab sampling.

sampling is not random. On the contrary regular and highly structured schemes are preferred since they are simpler to implement.

Now, let $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$ and $\underline{\boldsymbol{Y}}_1 \in \mathbb{R}^{I_1 \times J_1 \times K}$, $\underline{\boldsymbol{Y}}_2 \in \mathbb{R}^{I \times J \times K_2}$ as defined in (5.1). Then it holds:

$$\underline{\boldsymbol{Y}}_1 = [\![\boldsymbol{A}\left(\mathcal{S}_r, :\right), \boldsymbol{B}\left(\mathcal{S}_c, :\right), \boldsymbol{C}]\!] = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!] \tag{5.3a}$$

$$\underline{\boldsymbol{Y}}_2 = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}\left(\mathcal{S}_f, :\right)]\!] = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!] \tag{5.3b}$$

Identifiability of the model is established, under the conditions of the following theorem:

Theorem 5.1. *Let $\underline{\boldsymbol{X}} \in \mathbb{R}^{I \times J \times K}$, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. Assume that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ satisfy the equations in* (5.3)*. Then, $\boldsymbol{A}^\star = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1$, $\boldsymbol{B}^\star = \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2$, and $\boldsymbol{C}^\star = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3$, where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}_i$ is a full rank diagonal matrix such that $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$, provided that $2F + 2 \leq k_{\boldsymbol{A}^\star} + k_{\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3\boldsymbol{C}^\star}$ and $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ has full column rank.*

The proof of Theorem 5.1 is similar to the proof of Theorem 3.1 and therefore omitted. The

main insight is that if $\underline{\boldsymbol{Y}}_2$ admits a unique CPD, under Theorem 2.3, one can identify $\boldsymbol{A}$, $\boldsymbol{B}$ up to common permutation and scaling. Then $\boldsymbol{C}$ can be obtained from $\underline{\boldsymbol{Y}}_1$, via a linear system of equations, if $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ has full column rank.

5.2.2 Fiber sampling in multiple modes

We also propose a fiber sampling mechanism in multiple modes which can further reduce the complexity of computing the CPD. In particular, fiber samples are taken along different modes of the tensor, i.e. rows, columns and fibers are jointly sampled from $\underline{\boldsymbol{X}}$, as illustrated in Fig. 5.2. Following similar analysis as before we deduce:

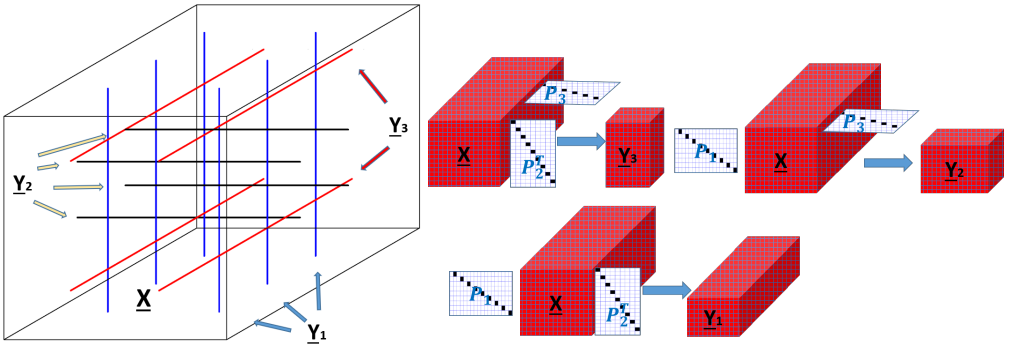


Figure 5.2: Multi-mode fiber sampling.

$$\underline{\boldsymbol{Y}}_1 \in \mathbb{R}^{I_1 \times J_1 \times K} = \underline{\boldsymbol{X}}\left(\mathcal{S}_r, \mathcal{S}_c, :\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 = [\![\boldsymbol{A}\left(\mathcal{S}_r, :\right), \boldsymbol{B}\left(\mathcal{S}_c, :\right), \boldsymbol{C}]\!]$$
$$= [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!] \tag{5.4a}$$
$$\underline{\boldsymbol{Y}}_2 \in \mathbb{R}^{I_2 \times J \times K_2} = \underline{\boldsymbol{X}}\left(\mathcal{S}_r, :, \mathcal{S}_f\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_3 \boldsymbol{P}_3 = [\![\boldsymbol{A}\left(\mathcal{S}_r, :\right), \boldsymbol{B}, \boldsymbol{C}\left(\mathcal{S}_f, :\right)]\!]$$
$$= [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!] \tag{5.4b}$$
$$\underline{\boldsymbol{Y}}_3 \in \mathbb{R}^{I \times J_3 \times K_3} = \underline{\boldsymbol{X}}\left(:, \mathcal{S}_c, \mathcal{S}_f\right) = \underline{\boldsymbol{X}} \times_2 \boldsymbol{P}_2 \times_3 \boldsymbol{P}_3 = [\![\boldsymbol{A}, \boldsymbol{B}\left(\mathcal{S}_c, :\right), \boldsymbol{C}\left(\mathcal{S}_f, :\right)]\!]$$
$$= [\![\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!] \tag{5.4c}$$

As far as identifiability is concerned, we have the following theorem:

Theorem 5.2. *Let* $\underline{\boldsymbol{X}} \in \mathbb{R}^{I \times J \times K}$*, with CPD* $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$*. Assume that* $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ *satisfy the equations in* (5.4)*. Then,* $\boldsymbol{A}^\star = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1$*,* $\boldsymbol{B}^\star = \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2$*, and* $\boldsymbol{C}^\star = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3$*, where* $\boldsymbol{\Pi}$ *is a*

permutation matrix and $\boldsymbol{\Lambda}_i$ *is a full rank diagonal matrix such that* $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$*, provided that* $2F+2 \leq k_{\boldsymbol{P}_1\boldsymbol{A}^\star} + k_{\boldsymbol{P}_2\boldsymbol{B}^\star} + k_{\boldsymbol{C}^\star}$ *and* $\boldsymbol{P}_3\boldsymbol{C}^\star \odot \boldsymbol{P}_2\boldsymbol{B}^\star$*,* $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ *have full column rank, or* $2F+2 \leq k_{\boldsymbol{P}_1\boldsymbol{A}^\star} + k_{\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3\boldsymbol{C}^\star}$ *and* $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$*,* $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_3\boldsymbol{C}^\star$ *have full column rank, or* $2F+2 \leq k_{\boldsymbol{A}^\star} + k_{\boldsymbol{P}_2\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3\boldsymbol{C}^\star}$ *and* $\boldsymbol{P}_3\boldsymbol{C}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$*,* $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ *have full column rank.*

In a nutshell, the above theorem states that the multi-mode fiber sampling model is identifiable if one of the subsampled tensors admits a unique CPD, under Theorem 2.3. The other two subtensors do not need to admit unique CPD's as long as they satisfy certain full column rank conditions. For example, factor $\boldsymbol{C}$ can be identified from the CPD of $\underline{\boldsymbol{Y}}_1$. Then $\boldsymbol{B},\ \boldsymbol{A}$ are computed from $\underline{\boldsymbol{Y}}_2,\ \underline{\boldsymbol{Y}}_3$ respectively, as solutions to linear system of equations.

5.3 Algorithmic framework

The first part of our approach selects an appropriate mechanism, which samples the given tensor, such that the CPD identifiability is maintained. In this section we develop an algorithmic framework which exploits the sampling pattern and reduces the computational and memory complexity of the CPD problem.

A three step approach is being followed for both sampling mechanisms.

Case of slab-fiber sampling: In the first step the CPD of $\underline{\boldsymbol{Y}}_2$ is computed and factors $\boldsymbol{A},\ \boldsymbol{B}$ are obtained. Then factor $\boldsymbol{C}$ is computed as the solution of the following linear system:

$$\boldsymbol{Y}_1^{(3)} = (\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{C}^T \tag{5.5}$$

Finally the following estimator is employed:

$$\text{minimize}_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}}\ \|\underline{\boldsymbol{Y}}_1 - [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2 + \|\underline{\boldsymbol{Y}}_2 - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!]\|_F^2\,, \tag{5.6}$$

Problem 5.6 is non-convex and NP-hard in general. To handle it we employ a *block coordinate descent (BCD)* approach with exact line search (perform a few gradient updates for each factor), which admits lightweight computations.

Case of multi-mode fiber sampling: The first step computes the CPD of $\underline{\boldsymbol{Y}}_1$ which obtains factor $\boldsymbol{C}$. Note that for multi-mode fiber sampling the CPD of $\underline{\boldsymbol{Y}}_2$ or $\underline{\boldsymbol{Y}}_3$, can be computed instead, with similar analysis. Step 2 computes the remaining factors, e.g. $\boldsymbol{A},\ \boldsymbol{B}$, as solutions to

the following system of linear equations:

$$\boldsymbol{Y}_2^{(2)} = (\boldsymbol{P}_3\boldsymbol{C} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{B}^T \tag{5.7a}$$

$$\boldsymbol{Y}_3^{(1)} = (\boldsymbol{P}_3\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})\boldsymbol{A}^T \tag{5.7b}$$

Finally, step 3 solves the following problem as before:

$$\begin{aligned}\text{minimize}_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}} \|\underline{\boldsymbol{Y}}_1 - [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2 + \|\underline{\boldsymbol{Y}}_2 - [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2 \\ + \|\underline{\boldsymbol{Y}}_3 - [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2\end{aligned} \tag{5.8}$$

Note that step 3 is crucial to obtain accurate solutions, especially on problems with real-noisy data. The FIber-Slab Tensor sampling algorithm (FIST) is presented in Algorithm 5.1.

Algorithm 5.1 FIST

Input: $\underline{\boldsymbol{X}}$,$F$.
Select sampling mechanism
Sample $\underline{\boldsymbol{X}}$ and generate $\underline{\boldsymbol{Y}}_1$, $\underline{\boldsymbol{Y}}_2$, $\underline{\boldsymbol{Y}}_3$.
Case Slab-fiber sampling:
1) $\boldsymbol{A}, \boldsymbol{B} \leftarrow$ CPD($\underline{\boldsymbol{Y}}_2$)
2) $\boldsymbol{C} \leftarrow$ solve (5.5).
3) If $\|\underline{\boldsymbol{X}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F >$ threshold:
Solve (5.6) using BCD with exact line search.
Case multimode fiber sampling:
1) $\boldsymbol{C} \leftarrow$ CPD($\underline{\boldsymbol{Y}}_1$)
2) $\boldsymbol{A}, \boldsymbol{B} \leftarrow$ solve (5.7).
3) If $\|\underline{\boldsymbol{X}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F >$ threshold:
Solve (5.8) using BCD with exact line search.

5.4 Simulations

In this section we showcase the effectiveness of the proposed framework with simulated experiments involving synthetically generated and real tensors. All simulations are performed in MATLAB on a Linux server with 8 3.6GHz cores and 32GB RAM.

The baseline algorithms used for comparison are:

`CPD`: The CPD of the original tensor $\underline{\boldsymbol{X}}$ is computed using Tensorlab's CPD command [166]. The stopping criterion is maximum number of iterations equal to 50, which empirically are

sufficient to give a good CPD fit.

`Randomized Block Sampling` (`RBS`) [167]: Tensorlab's implementation is being used and the algorithm is tested for different block sizes.

`Paracomp` [145]: Author's implementation is being used with three anchor rows between the compressed factors to reconcile for permutation and scaling mismatches. The CPD of the compressed tensors is performed with 50 iterations of Tensorlab's algorithm.

$\texttt{FIST}_1$, $\texttt{FIST}_2$: The two proposed approaches for slab-fiber and multi-fiber sampling respectively. We run step 1 with 50 iterations of Tensorlab's algorithm and the CPD stopping criterion is maximum number of iterations equal to 50. The threshold is set equal to $10^{-2}\|\underline{\boldsymbol{X}}\|_F$ and the stopping criterion for step 3 is maximum number of iterations equal to 5.

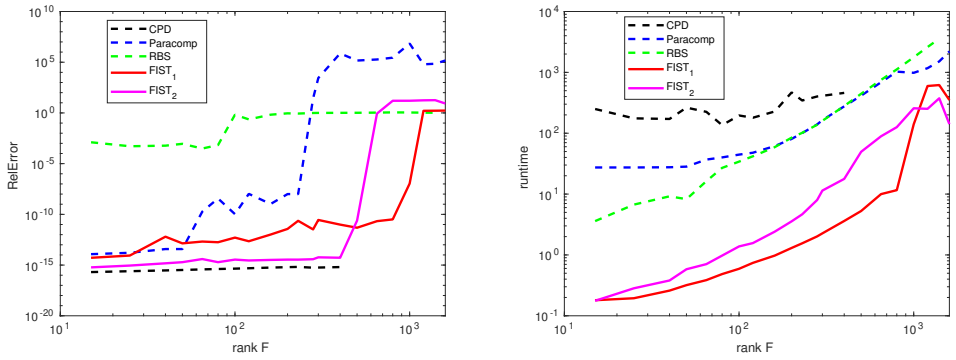
To assess the performance of each algorithm we measure the CPD relative error defined as:

$$\texttt{RelError} = \frac{\|\underline{\boldsymbol{X}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F}{\|\underline{\boldsymbol{X}}\|_F},$$

where the subscript F is used to denote the Frobenius norm of a tensor. We also measure the `runtime` of each algorithm.

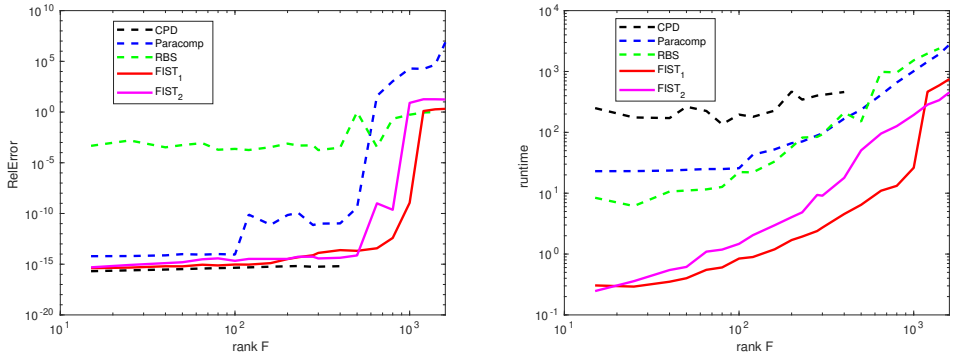
5.4.1 Synthetic experiments

The first set of experiments uses synthetically generated third-order tensors. In particular, we generate tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{1000\times1000\times1000}$ by randomly drawing the CPD factors $\boldsymbol{A} \in \mathbb{R}^{1000\times F}$, $\boldsymbol{B} \in \mathbb{R}^{1000\times F}$, $\boldsymbol{C} \in \mathbb{R}^{1000\times F}$ from a zero-mean unit-variance Gaussian distribution and synthesize the tensor as $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. We vary the rank F from 15 to 1600 and record the `RelError` and `runtime` for all the competing methods. Two scenarios are considered. In the first, the sampling/compression ratio $r = \frac{\#\text{measurements}}{IJK}$, for our method as well as `Paracomp`, is in the order of 10^{-3} and for the second in the order of 10^{-2}. Then for $\texttt{FIST}_1$ $K_2 = 2, 5$ for the two scenarios and $I_1 = J_1$ are chosen such that $I_1 J_1 > F + 10$. Regarding $\texttt{FIST}_2$ $I_1 = I_2 = J_1 = J_3 = 40, 50$ and $K_2 = K_3$ are chosen such that $I_1 K_2 > F + 10$. As far as `Paracomp` is concerned, the compressed subtensors, for the two scenarios, are chosen to be of size $50 \times 50 \times 50$ or $100 \times 100 \times 100$ and their number is $n = 22$ and $n = 11$ respectively, so that the final system is overdetermined. The block sizes in `RBS` are chosen equal to `Paracomp` for fair comparison. The performance of the competing methods for the two scenarios is presented in Fig. 5.3, 5.4 respectively. In terms of `RelError` $\texttt{FIST}_1$, $\texttt{FIST}_2$ work the best and



(a) `RelError` (b) `runtime`

Figure 5.3: Scenario 1



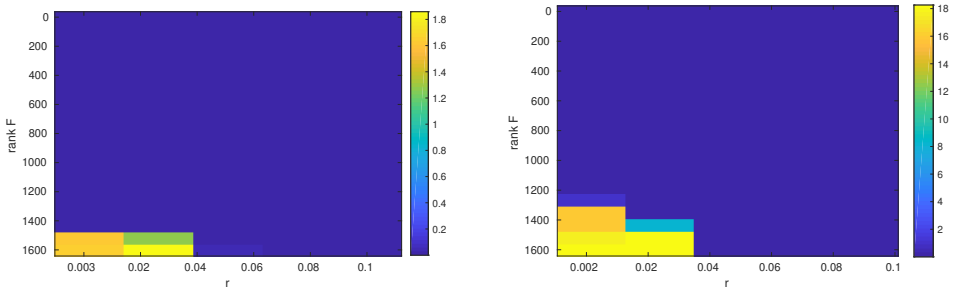
(a) `RelError` (b) `runtime`

Figure 5.4: Scenario 2

`Paracomp` comes second (for small ranks). As far as `runtime` is concerned $\texttt{FIST}_1$ is the fastest, while $\texttt{FIST}_2$ comes second. Note that both of them are at least an order of magnitude faster than the competing algorithms. The `RBS` algorithm exhibits a stable performance. We should also mention the direct CPD on the full tensors runs out of memory for rank greater than 500. We also vary the sampling ratio r from 0.002 to 0.1 for our proposed methods. Fig 5.5 presents the `RelError` for $\texttt{FIST}_1$ and $\texttt{FIST}_2$ as a function of F and r. The results show that the proposed methods work well for a wide range of ranks and sampling ratios.

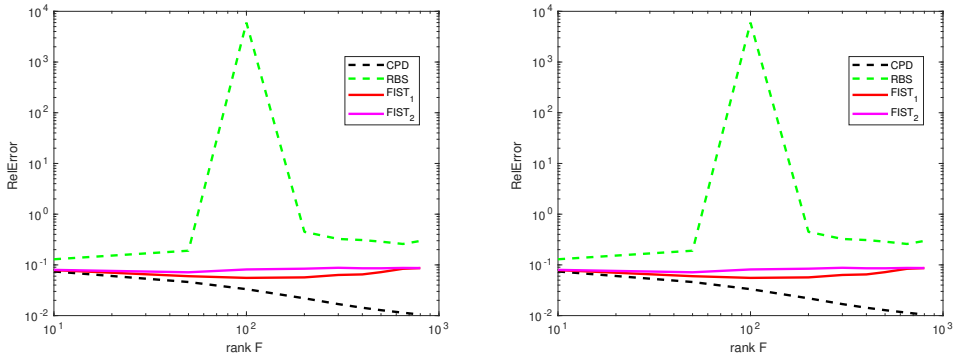
5.4.2 Real experiments

Finally we test the proposed approach with real data tensors. To this end we use the Cuprite hyperspectral image (HSI) from the AVIRIS platform [162], which is represented as a third order



(a) $\texttt{FIST}_1$ (b) $\texttt{FIST}_2$

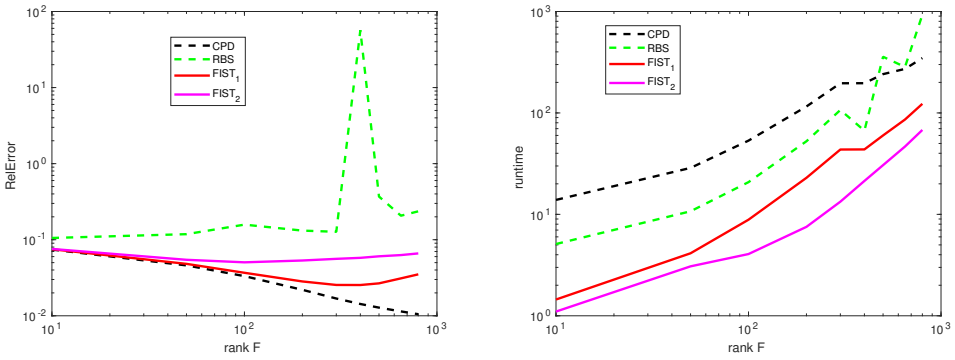
Figure 5.5: F vs r



(a) `RelError` (b) `runtime`

Figure 5.6: Real scenario 1

tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{512\times614\times187}$. Note that in HSI's factor $\boldsymbol{C}$ is generally ill-conditioned, due to the low rank matrix structure HSI's admit. In particular the condition number of Cuprite HSI for different ranks ranges from 10^4 to 10^8. We vary the rank from 10 to 800 and consider again two scenarios: In the first $I_1 = J_1 = 40$, $K_2 = 2$ for $\texttt{FIST}_1$ and $I_1 = I_2 = J_1 = J_3 = K_2 = K_3 = 50$ for $\texttt{FIST}_2$, whereas the blocksize of `RBS` is $10 \times 10 \times 10$. In the second scenario $I_1 = J_1 = 40$, $K_2 = 5$ for $\texttt{FIST}_1$, $I_1 = I_2 = J_1 = J_3 = 100$, $K_2 = K_3 = 50$ for $\texttt{FIST}_2$ and `RBS` block is $20 \times 20 \times 20$. Note that for `RBS` block sizes greater than 30 the runtime is worse than `CPD`. The performance of `Paracomp` for all sizes and ranks was giving `RelError` greater than 10 and thus is omitted. The reason is that with real noisy data reconciling for permutation and scaling mismatches becomes very cumbersome. The results are presented in Fig. 5.6, 5.7. Same conclusions can be derived again. The proposed $\texttt{FIST}_1$, $\texttt{FIST}_2$ are faster and more accurate than the competitors.



(a) `RelError` (b) `runtime`

Figure 5.7: Real scenario 2

5.5 Conclusion

In this chapter we studied the task of computing the CPD of large-scale tensors. We proposed two sampling mechanisms that operate on different modes of the tensor. The sampling is regular and does not need to follow any stochasticity. We also established the identifiability of the proposed model and developed an efficient algorithmic framework to handle the problem. Simulations with synthetic and real experiments showcase the effectiveness of the approach.

Chapter 6

GAGE: Geometry Preserving Attributed Graph Embeddings

Node representation learning is the task of extracting concise and informative feature embeddings of certain entities that are connected in a network. Many real world network datasets include information about both node connectivity and certain node attributes, in the form of features or time-series data. Modern representation learning techniques utilize both connectivity and attribute information of the nodes to produce embeddings in an unsupervised manner. In this context, deriving embeddings that preserve the geometry of the network and the attribute vectors would be highly desirable, as they would reflect both the topological neighborhood structure and proximity in feature space. While this is fairly straightforward to maintain when only observing the connectivity or attributed information of the network, preserving the geometry of both types of information is challenging. A novel tensor factorization approach for node embedding in attributed networks that preserves the distances of both the connections and the attributes is proposed in this chapter, along with an effective and lightweight algorithm to tackle the learning task. Judicious experiments with multiple state-of-art baselines suggest that the proposed algorithm offers significant performance improvements in node classification and link prediction tasks.

6.1 Prior Art

A plethora of methods have been proposed to perform node embedding. Early works approached the node representation learning task using only the connectivity information of the network. A number of them focused on properly defining a similarity measure on the connectivity information and performing matrix factorization on it [4, 22, 33, 123, 132, 140, 157, 161, 183]. Random walks have also been successfully employed to generate node embeddings, e.g., [62,128]. More recently, the focus of research has shifted towards generating embeddings for attributed networks. The work in [183] generalizes deepwalk [128] to the case where attributes are available, while [72] performs label-informed attributed node representation learning in a semi-supervised setting. Neural for network tasks have also gained significant attention lately. In particular, graph convolutional neural networks and graph auto-encoders are very popular for attributed node embedding [34,91,92,164,173]. Works have also been proposed to perform inductive embedding, e.g., [63] where a graph convolutional network is trained with multiple graphs. Finally, the work in [9] employs a tensor decomposition model and jointly factors the conventional adjacency along with a k-nearest neighbor matrix of the attributes.

6.2 Problem Statement

The problem can be informally stated as follows:

- **Given:** the connectivity and attribute information of network nodes.
- **Produce:** Low dimensional node representations that preserve both the connectivity and attribute geometry.

We begin the discussion with the definition of node embedding. Let $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$ be a directed or undirected graph, with $\mathcal{V}$ being the set of $N = |\mathcal{V}|$ nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ being the set of edges. We are also given a set of attributes $\boldsymbol{A}$ for each node. Node embedding aims to map each node to a vector in $F-$dimensional Euclidean space. Formally, the node embedding task seeks for a function $f(\cdot) : \mathcal{G}, \boldsymbol{A} \rightarrow \mathbb{R}^{N \times F}$, where $F \ll N$. The node embeddings can be represented by matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N]^T$, where each row that contains the F-dimensional embedding of each node.

6.2.1 Related work

Recent work [9] proposed building a tensor $\underline{\boldsymbol{X}}$ whose first frontal slab $\boldsymbol{X}_1$ is the network adjacency matrix, while its second frontal slab $\boldsymbol{X}_2$ is the an attribute adjacency, obtained by computing the set of k nearest neighbors [129] of each node in attribute space. In other words, the attributes of a given node are viewed as a vector in Euclidean space, and the k closest attribute vectors of other nodes in the network are used to define the neighbors of the given node. The number of nearest neighbors is a parameter that needs to be tuned. A second adjacency matrix is produced this way, which however is not necessarily symmetric (even if the original network adjacency is). Joint analysis of these two adjacency matrices yields embeddings that reflect both pieces of information – but are not geometry-preserving, because (approximately) reproducing these adjacency matrices has no geometric motivation / interpretation.

In this work we propose a principled formulation that directly aims to produce an embedding that can reproduce the distances between nodes in terms of their network adjacency and in terms of their attributes. We find common latent dimensions that explain both sets of distances. With proper weighting, the resulting embedding vectors reproduce the adjacency distances; with another weighting, they reproduce the attribute distances (and these weights are a by-product of our analysis). Either way, the latent dimensions are derived from (and reflect) both sets of distances. This is why we call the approach *geometry preserving*. Depending on the downstream task, different weighting schemes would be more appropriate. Our formulation draws from multi dimensional scaling (MDS), which is briefly reviewed next.

6.2.2 Multi dimensional scaling

MDS is a distance-preserving mapping, visualization, and embedding tool [44, 98, 137, 159]. Given an $N \times N$ matrix $\boldsymbol{D}$ of distances between N entities, MDS seeks to find N points in low-dimensional space (typically 2- or 3-dimensional, for visualization purposes) that approximately exhibit the given distances. Various distances (and pseudo-distances) can be used for MDS. The most popular is the Euclidean distance, leading to the classical MDS, but there exist non-metric versions of MDS which seek to preserve ordering as opposed to distances [98]. We next briefly review classical MDS. Let $\boldsymbol{D}^{(2)} \in \mathbb{R}^{N \times N}$ be the matrix of squared distances between N entities, with $\boldsymbol{D}^{(2)}(i,j)$ being the squared distance between entity i and entity j. Now let $\boldsymbol{e}_i$ be the vector

representation of entity i in a low F-dimensional Euclidean space. Then it holds that:

$$\boldsymbol{D}^{(2)}(i,j) = \|\boldsymbol{e}_i - \boldsymbol{e}_j\|^2 = \|\boldsymbol{e}_i\|^2 + \|\boldsymbol{e}_j\|^2 - 2\boldsymbol{e}_i^T\boldsymbol{e}_j \tag{6.1}$$

Since the objective is to learn the $\{\boldsymbol{e}_i\}_{i=1}^N$ we would like to end up with an expression that ignores the squared norms $\|\boldsymbol{e}_i\|^2, \|\boldsymbol{e}_j\|^2$ and will be easy to factor. In this direction we observe that:

$$\boldsymbol{D}^{(2)} = \boldsymbol{g}\mathbf{1}^T + \mathbf{1}\boldsymbol{g}^T - 2\boldsymbol{E}\boldsymbol{E}^T, \tag{6.2}$$

where $\boldsymbol{g} = \left[\boldsymbol{e}_1^T\boldsymbol{e}_1, \ldots, \boldsymbol{e}_N^T\boldsymbol{e}_N\right]^T$. Double centering both sides yields:

$$\begin{aligned}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{D}^{(2)}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{g}\mathbf{1}^T\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) + \\ \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathbf{1}\boldsymbol{g}^T\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) - \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)2\boldsymbol{E}\boldsymbol{E}^T\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right),\end{aligned} \tag{6.3}$$

which is equivalent to

$$-\frac{1}{2}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{D}^{(2)}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) = \boldsymbol{E}\boldsymbol{E}^T, \tag{6.4}$$

since $\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathbf{1} = 0$ and we can assume without loss of generality that matrix $\boldsymbol{E}$ is already centered. The solution for $\boldsymbol{E}$ is given by

$$\boldsymbol{E} = \boldsymbol{U}\sqrt{\boldsymbol{\Lambda}_F}, \tag{6.5}$$

where $\boldsymbol{U} \in \mathbb{R}^{N\times F}$ is the matrix of F principal eigenvectors and $\boldsymbol{\Lambda}_F \in \mathbb{R}^{F\times F}$ a diagonal matrix with the F principal eigenvalues of $-\frac{1}{2}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{D}^{(2)}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$. In the non-ideal case where $\boldsymbol{D}^{(2)}$ is inexact, assuming that the left hand side of (6.4) remains (or is projected to be) positive semidefinite, (6.5) gives the best vector representation of the entities in an F-dimensional space *after double-centering*, albeit that is not optimal from the viewpoint of preserving the original distances. For the latter, we need to resort to iterative algorithms that minimize a suitable cost (or *stress*) function, but that is often not necessary in practice.

MDS has also been generalized to the case where more than one distance matrices are available for a set of entities [37]. For example, the entities could be a set of N different products and K individuals are asked to rate their similarity or dissimilarity. This results in K different

$N \times N$ distance matrices for the N products. To be more precise let $\boldsymbol{D}_k^{(2)} \in \mathbb{R}^{N \times N}$ be the k-th given distance matrix. Three-way MDS forms a third-order tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{N \times N \times K}$ as:

$$\underline{\boldsymbol{X}}(:,:,k) = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) \boldsymbol{D}_k^{(2)} \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) \tag{6.6}$$

and performs CPD of $\underline{\boldsymbol{X}}$ to find a joint F-dimensional representation of the entities.

6.2.3 GAGE: Geometry preserving Attributed Graph Embeddings

In the previous section we introduced the task of unsupervised node embedding. The objective of this task is to map each node of the network to a low dimensional vector representation in the Euclidean space. It is desirable that the low dimensional embedding contains as much connectivity and attribute information as possible and progress in this direction is the key to successful embeddings.

In this section, motivated by the benefits of MDS, we propose a novel unsupervised node embedding scheme that works with attributed networks. The proposed node embedding scheme attempts to preserve the network geometry inferred both from connectivity and attribute information. Furthermore, the node embeddings are unique. Note that uniqueness is a fundamental property than each embedding should enjoy. It offers a unique representation of each node, which is necessary for any form of interpretability and also guarantees that the embedding is permutation invariant. In other words any permuted version of the adjacency yields the same embedding for each node. Finally the proposed representation model is flexible in the sense that it can handle both directed and undirected graphs and does not require connectivity and attributed information for every node. In other words embeddings can be produced for nodes with either missing connectivity information or missing attributes.

Traditional MDS starts from a distance matrix and looks for vector representations of the nodes. In our setting we are given the adjacency representation of each node along with a vector of attributes. The obvious approach would be to try and learn low-dimensional node embeddings directly from the high-dimensional graph and attribute representation of each node. However, since our objective is the produced embeddings to preserve the network geometry in terms of Euclidean distances, we propose to follow a different route. In particular, given the adjacency and the attributes of the network we compute distance matrices, one for the connectivity information and another for the attribute information. This transformation from

adjacency and attributes to connectivity and attribute distances is the key to our proposed geometry preserving embeddings. Then we decompose the tensor of distances, using the CPD model, and produce the low-dimensional embeddings. As we see later in the section, the produced embeddings, which are formed from the CPD factors, can reproduce both the connectivity and attribute distances. Note that, from a computational viewpoint, instantiating the Euclidean distance matrices of connectivity and attributes might be prohibitive, since it destroys the sparsity structure. Interestingly, there is a elegant way to overcome it.

In order to facilitate the analysis let $\boldsymbol{S}_{\mathcal{G}} \in \{0,1\}^{N\times N}$ denote the adjacency matrix of graph $\mathcal{G}$ and $\boldsymbol{A} \in \mathbb{R}^{N\times d}$ be the matrix the contains in row i d attributes or features of vertex i. Also let $\boldsymbol{Y}^1 = \boldsymbol{S}_{\mathcal{G}}$ and $\boldsymbol{Y}^2 = \boldsymbol{A}$. Taking a closer look at equation (6.3) we observe that double centering the matirx of Euclidean distances between the rows of $\boldsymbol{Y}^1$ or $\boldsymbol{Y}^2$ is equal to double centering $\boldsymbol{Y}^1\boldsymbol{Y}^{1^T}$ or $\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}$. This is due to the fact that $\boldsymbol{Y}^1$ or $\boldsymbol{Y}^2$ contain the generating vectors of the distance matrices and equation (6.3) always holds. We now transform the adjacency and attribute to distance matrices:

$$\boldsymbol{X}^1 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^1\boldsymbol{Y}^{1^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right), \tag{6.7}$$

$$\boldsymbol{X}^2 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right). \tag{6.8}$$

Note that $\boldsymbol{X}^1(i,j)$ denotes the squared Euclidean distance (after double centering) between $\boldsymbol{S}_G(i,:)$ and $\boldsymbol{S}_G(j,:)$, i.e., two rows of the adjacency matrix. Also $\boldsymbol{X}^2(i,j)$ denotes the squared Euclidean distance (after double centering) between $\boldsymbol{A}(i,:)$ and $\boldsymbol{A}(j,:)$, i.e., two attributed information of different nodes. It is important to mention that in most applications $\boldsymbol{S}_{\mathcal{G}}$ and $\boldsymbol{A}$ are sparse matrices which facilitates storage and computation requirements. Double centering these matrices automatically yields dense matrices. However, as we will see next our approach doesn't instantiate the dense $\boldsymbol{X}^1$ and $\boldsymbol{X}^2$ but works with sparse $\boldsymbol{Y}^1$ and $\boldsymbol{Y}^2$, which is crucial to keep the computational and memory complexity of the algorithm low.

To compute the node embeddings of the attributed network, we propose to employ the following optimization scheme:

$$\min_{\boldsymbol{U},\boldsymbol{\Lambda}_1,\boldsymbol{\Lambda}_2} \|\boldsymbol{X}^1 - \boldsymbol{U}\boldsymbol{\Lambda}_1\boldsymbol{U}^T\|_F^2 + \|\boldsymbol{X}^2 - \boldsymbol{U}\boldsymbol{\Lambda}_2\boldsymbol{U}^T\|_F^2, \tag{6.9}$$

where $\boldsymbol{U} \in \mathbb{R}^{N \times F}$ and $\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2$ are real and positive valued $F \times F$ diagonal matrices. Problem (6.9) is the rank F CPD of tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{N \times N \times 2}$, with frontal slabs $\underline{\boldsymbol{X}}(:,:,1) = \boldsymbol{X}^1$ and $\underline{\boldsymbol{X}}(:,:,2) = \boldsymbol{X}^2$. The CPD model for $\underline{\boldsymbol{X}}$ takes the form:

$$\underline{\boldsymbol{X}} = [\![\boldsymbol{U}, \boldsymbol{U}, \boldsymbol{C}]\!], \quad \boldsymbol{C}(i,:)^T = \text{diag}\left(\boldsymbol{\Lambda}_i\right), \; i = 1, 2, \tag{6.10}$$

where $\text{diag}\left(\boldsymbol{\Lambda}_i\right)$ is the diagonal vector of $\boldsymbol{\Lambda}_i$. The proposed embedding for vertex i is :

$$\boldsymbol{e}_i = \boldsymbol{E}(i,:)^T = \text{diag}\left(\sqrt{\lambda \boldsymbol{C}(:,1)^T + (1-\lambda)\boldsymbol{C}(:,2)^T}\right) \boldsymbol{U}(i,:)^T, \tag{6.11}$$

where $\text{diag}\left(\sqrt{\lambda \boldsymbol{C}(:,1)^T + (1-\lambda)\boldsymbol{C}(:,2)^T}\right)$ gives the diagonal matrix of vector $\sqrt{\lambda \boldsymbol{C}(:,1)^T + (1-\lambda)\boldsymbol{C}(:,2)^T}$. Note that the $0 \leq \lambda \leq 1$ parameter balances the contribution of each distance measure (connectivity or attribute) in the final embedding. For $\lambda = 1$ the focus is completely on the connectivity distances, whereas for $\lambda = 1$ the emphasis is on the attribute distances.

Invoking the uniqueness properties o the CPD (see Theorems 2.1, 2.2 or 2.3 for details) we have shown the following result:

Result 6.1. *If tensor $\underline{\boldsymbol{X}}$ has indeed low-rank, F, there exist vectors in F dimensional space that generate the given sets of distances (with appropriate weights). Then the* GAGE *embeddings for the correct F are unique, permutation invariant and will exactly reproduce both sets of distances for $\lambda = 0$ and $\lambda = 1$.*

The above result also implies that embeddings of dimension less than F cannot reconstruct the set of distances and embeddings of dimension larger than F are not unique.

6.3 Algorithmic framework

In this section we discuss the algorithmic aspects of our approach.

6.3.1 The GAGE algorithm

The computation of the proposed node embeddings boil down to solving the problem in (6.9). This is a CPD problem of an $N \times N \times 2$ tensor with a special sparsity structure on the frontal slabs.

CPD computation is a non-convex optimization problem and in general NP-hard. However, exact CPD can be reduced to eigenvalue decomposition (EVD) in certain cases – notably when tensor rank is low enough [48,136]. Such an approach is not guaranteed to produce the optimal solution, but it often works well in practice, and it also serves as good initialization for more sophisticated optimization approaches. Developing a computationally efficient algebraic initialization approach to tackle the problem in (6.9) is therefore pivotal to the proposed algorithm. This is GAGE−EVD, which is summarized in Algorithm 1. The first step of the approach is to form the doubly centered frontal slabs. Note that instantiating $\boldsymbol{X}^1,\ \boldsymbol{X}^2$ is not required; GAGE−EVD exploits sparsity and the special problem structure to mitigate memory and complexity requirements, as shown in the Appendix. The next step is to compute the F principal eigenvectors $\boldsymbol{V}$ of ${\boldsymbol{X}^1}^T\boldsymbol{X}^1 + {\boldsymbol{X}^2}^T\boldsymbol{X}^2$. Towards this end. we employ the orthogonal iterations method [59] which also exploits the special sparsity structure to enable lightweight computations. The details can be found in the Appendix. Finally, we form $\boldsymbol{S}_1 = \boldsymbol{V}^T\boldsymbol{X}^1\boldsymbol{V},\ \boldsymbol{S}_2 = \boldsymbol{V}^T\boldsymbol{X}^2\boldsymbol{V}$ which are dense but small $(F \times F)$ matrices and compute the eigenvalue decomposition of $\boldsymbol{S}_2\boldsymbol{S}_1^{-1}$. Then $\boldsymbol{U}$ is computed as $\boldsymbol{U}^T = \tilde{\boldsymbol{U}}^{-1}\boldsymbol{S}_1$. In terms of computational complexity, the main bottleneck of GAGE−EVD is computing the EVD of ${\boldsymbol{X}^1}^T\boldsymbol{X}^1 + {\boldsymbol{X}^2}^T\boldsymbol{X}^2$. Using the orthogonal iterations method, this EVD can be computed efficiently in $\mathcal{O}(NF^2)$ flops. The remaining operations involve $F \times F$ matrices and are computationally light. Detailed description of computational complexity and memory requirements is given in the Appendix.

After computing an initial estimate of matrix $\boldsymbol{U}$, we feed it to the main GAGE algorithm, which is summarized in Algorithm 2. To tackle the problem in (6.9) GAGE follows an alternating least squares approach, with the first two factors $\boldsymbol{U}, \boldsymbol{U}'$ not constrained to be equal. In each update, we fix two factors and solve for the remaining one. We repeat this procedure in an alternating fashion. The update for each step is a linear system of equations and can be solved efficiently without instantiating the dense tensor $\underline{\boldsymbol{X}}$ or any of the Khatri-Rao products, i.e., $(\boldsymbol{C} \odot \boldsymbol{U}'),\ (\boldsymbol{C} \odot \boldsymbol{U}'),\ (\boldsymbol{U}' \odot \boldsymbol{U})$. The details are presented in the Appendix. Note that due to the algebraic initialization, the GAGE algorithm converges in only a few steps (usually fewer than 10) in our experiments.

Algorithm 6.1 GAGE-EVD

Input: $\boldsymbol{Y}^1 = \boldsymbol{S}_{\mathcal{G}}, \boldsymbol{Y}^2 = \boldsymbol{A},\ F.$

Output: $\boldsymbol{U}.$

$\boldsymbol{X}^1 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^1\boldsymbol{Y}^{1^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right);$

$\boldsymbol{X}^2 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right);$

$\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T \leftarrow \text{EVD}\Big(\boldsymbol{X}^{1^T}\boldsymbol{X}^1 + \boldsymbol{X}^{2^T}\boldsymbol{X}^2, F\Big);$

$\boldsymbol{S}_1 = \boldsymbol{V}^T\boldsymbol{X}^1\boldsymbol{V},\ \boldsymbol{S}_2 = \boldsymbol{V}^T\boldsymbol{X}^2\boldsymbol{V};$

$\tilde{\boldsymbol{U}} \leftarrow \text{EVD}\left(\boldsymbol{S}_2\boldsymbol{S}_1^{-1}\right);$

$\boldsymbol{U}^T = \tilde{\boldsymbol{U}}^{-1}\boldsymbol{S}_1;$

Algorithm 6.2 GAGE

Input: $\boldsymbol{Y}^1 = \boldsymbol{S}_{\mathcal{G}}, \boldsymbol{Y}^2 = \boldsymbol{A},\ \boldsymbol{U}.$

Output: $\boldsymbol{E}.$

$\boldsymbol{X}^1 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^1\boldsymbol{Y}^{1^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right);$

$\boldsymbol{X}^2 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right);$

$\underline{\boldsymbol{X}}(:,:,1) = \boldsymbol{X}^1,\ \underline{\boldsymbol{X}}(:,:,2) = \boldsymbol{X}^2;$

$\boldsymbol{C} \leftarrow \text{solve } \boldsymbol{X^{(3)}} = (\boldsymbol{U} \odot \boldsymbol{U})\boldsymbol{C}^T;$

$\boldsymbol{U}' = \boldsymbol{U};$

repeat

  $\boldsymbol{U} \leftarrow \text{solve } \boldsymbol{X^{(1)}} = (\boldsymbol{C} \odot \boldsymbol{U}')\boldsymbol{U}^T;$

  $\boldsymbol{U}' \leftarrow \text{solve } \boldsymbol{X^{(2)}} = (\boldsymbol{C} \odot \boldsymbol{U})\boldsymbol{U}'^T;$

  $\boldsymbol{C} \leftarrow \text{solve } \boldsymbol{X^{(3)}} = (\boldsymbol{U}' \odot \boldsymbol{U})\boldsymbol{C}^T;$

until convergence

$\boldsymbol{E} = \boldsymbol{U}\ \text{diag}\left(\sqrt{\lambda\boldsymbol{C}(:,1)^T + (1-\lambda)\boldsymbol{C}(:,2)^T}\right);$

6.4 Experiments

In this section we demonstrate the performance of the proposed algorithmic framework and showcase its effectiveness in experiments with real attributed network data. All algorithms were implemented in Matlab or Python, and executed on a Linux server comprising 8 cores at 3.6GHz with 32GB RAM.

Table 6.1: Datasets

Dataset	# Vertices	# Edges	Attribute dimension	# Classes	Network Type	Feature Type
`Wikipedia`	2,405	23,192	4,973	19	Language	Text associated info
`WebKB`	877	2,776	1,703	5	Citation	Unique words
`BlogCatalog`	5,196	686,972	8,189	6	Social	Keywords

6.4.1 Data

In our experiments, we used the following real-world networks (see also Table 2).

- **`BlogCatalog`**. A social network of bloggers in BlogCatalog platform. Each blogger uses several keywords to describe their blogs. These keywords have been used as attributes for the node-bloggers. There are 6 different classes of bloggers according to the content of their blogs.
- **`WebKD`** [57]. A network of webpages from computer science departments categorized into 5 topics: faculty, student, project, course, other. The attributes dimension is a dictionary of words that appear in the webpages.
- **`Wikipedia`** [183]. A network of documents and their Wikipedia links. The documents are grouped into 19 classes and the attribute information is text related.

6.4.2 Baselines

Methods. Experiments were run using the following *unsupervised* embedding methods.

- **Deepwalk** [128]. `Deepwalk` generates truncated random walks from each node, to learn low dimensional representations of nodes using a SkiGram model. We set the number of walks per node $\gamma = 80$, walk length $t = 40$ and window size $w = 10$ as suggested in [128]. This method does not use the attributes, and it is not expected to work as well as the other methods that do. We include it here because it remains a strong contender when only the network adjacency is available, and as a means to gauge the improvement afforded by having access to the node attributes.
- **T-Pine** [9]. A tensor factorization based approach. The first frontal slab is the adjacency of the graph and the second frontal slab is the a k nearest neighbor matrix computed using

the distances between the node attributes. The k nearest neighbor parameter is set to $k = 8$ for `Wikipedia`, $k = 40$ for `WebKB` as suggested in [9] and $k = 50$ for `BlogCatalog`.

- **Graph-AE** [92]. A graph convolutional network (GCN) generalization for unsupervised node embedding. This approach uses a (GCN) encoder and a simple inner product decoder.
- **Graph-VAE** [92]. A variational auto encoder (VAE) alternative to `Graph-AE`. Both `Graph-AE` and `Graph-VAE` are trained using 200 epochs and 0.01 learning rate. The dimension of the hidden layer is twice the number of the embedding dimension. We also use 5% of the data for validation.
- **TADW** [183]. Text associated Deepwalk (TADW) employs a matrix factorization framework to learn network representations using the adjacency matrix as well as textual information features.
- **DGI** [164]. Deep Graph Infomax (DGI) uses a graph convolutional neural network architecture to learn node embeddings for attributed networks in an unsupervised manner. We train for maximum 1000 epochs using the code provided by the authors and set the 'patience' parameter equal to 20 and learning rate equal to 0.001, as suggested in [164].

6.4.3 Node classification

We first test the performance of the proposed `GAGE` along with the baselines in a node classification task. The procedure is divided in two steps. In the first step the algorithms learn the node embeddings in a unsupervised manner, i.e., without using label information. In the second step the labels along with the learned embeddings are split into training and testing sets. Then the training data are fed to a one-versus-all logistic regression classifier with l_2 norm regularization. We test 3 different training-testing splits, i.e., 0.9-0.1, 0.5-0.5, and 0.1-0.9 and run 10 shuffles for each split. To assess the performance of the competing algorithms we measure the average micro and macro F1 score for 2 different embedding dimensions. For the `GAGE` embeddings we set $\lambda = 0.8$. The results for the three different datasets are presented in Tables 6.2, 6.3, 6.4.

It is clear from the tables that the proposed `GAGE` significantly outperforms the baselines in both micro and macro F1 score, where `T-PINE` usually comes second. In the `Wikipedia` dataset there are instances where `T-PINE` is slightly better in micro F1 but `GAGE` is better in macro F1. Taking into consideration that the `Wikipedia` dataset consists of 19 classes

Table 6.2: Average score and standard deviation over 10 shuffles for `Wikipedia`

Algorithm	dimension	micro (0.9)	macro (0.9)	micro (0.5)	macro (0.5)	micro (0.1)	macro (0.1)
GAGE	64	$\mathbf{0.7402 \pm 0.0308}$	$\mathbf{0.5331 \pm 0.0239}$	$\mathbf{0.7303 \pm 0.0125}$	$\mathbf{0.5262 \pm 0.0217}$	$\mathbf{0.6309 \pm 0.0217}$	$\mathbf{0.423 \pm 0.0246}$
	128	0.7656 ± 0.0255	$\mathbf{0.5924 \pm 0.0337}$	0.736 ± 0.0104	$\mathbf{0.5802 \pm 0.0198}$	$\mathbf{0.649 \pm 0.0179}$	$\mathbf{0.4728 \pm 0.0261}$
T-PINE	64	0.6788 ± 0.0312	0.4039 ± 0.0158	0.6619 ± 0.0072	0.3949 ± 0.0047	0.5912 ± 0.0139	0.3535 ± 0.0042
	128	$\mathbf{0.766 \pm 0.0234}$	0.523 ± 0.0183	$\mathbf{0.745 \pm 0.009}$	0.5069 ± 0.0121	0.6364 ± 0.0081	0.4205 ± 0.0144
Deepwalk	64	0.6177 ± 0.0309	0.3632 ± 0.0213	0.6136 ± 0.0038	0.3736 ± 0.0119	0.5773 ± 0.0084	0.3415 ± 0.0126
	128	0.6236 ± 0.0333	0.362 ± 0.0175	0.614 ± 0.006	0.3731 ± 0.0111	0.5811 ± 0.0095	0.3444 ± 0.0126
Graph-AE	64	0.6759 ± 0.0314	0.4512 ± 0.0335	0.6481 ± 0.0117	0.4254 ± 0.0193	0.5669 ± 0.0075	0.3452 ± 0.0121
	128	0.6747 ± 0.0372	0.4327 ± 0.0346	0.6584 ± 0.0082	0.4287 ± 0.0203	0.5773 ± 0.01	0.3536 ± 0.0115
Graph-VAE	64	0.6283 ± 0.03	0.4108 ± 0.0297	0.6069 ± 0.009	0.3804 ± 0.0137	0.5592 ± 0.0112	0.3323 ± 0.0124
	128	0.67 ± 0.0394	0.436 ± 0.028	0.6404 ± 0.0119	0.4098 ± 0.0195	0.5742 ± 0.0107	0.3431 ± 0.0109
TADW	64	0.7008 ± 0.0258	0.4541 ± 0.0244	0.6990 ± 0.0142	0.4781 ± 0.0156	0.6160 ± 0.0121	0.3996 ± 0.0126
	128	0.7510 ± 0.0345	0.5378 ± 0.0373	0.7168 ± 0.0135	0.5170 ± 0.0224	0.6309 ± 0.0159	0.4221 ± 0.0218
DGI	64	0.5523 ± 0.0258	0.2806 ± 0.0095	0.4927 ± 0.0182	0.2271 ± 0.0124	0.3157 ± 0.0323	0.0741 ± 0.0139
	128	0.5299 ± 0.0254	0.252 ± 0.0139	0.4563 ± 0.0171	0.1825 ± 0.0119	0.2924 ± 0.0458	0.066 ± 0.0148

Table 6.3: Average score over 10 shuffles for WebKB

Algorithm	dimension	micro (0.9)	macro (0.9)	micro (0.5)	macro (0.5)	micro (0.1)	macro (0.1)
GAGE	64	$\mathbf{0.8852 \pm 0.0375}$	$\mathbf{0.7588 \pm 0.0506}$	$\mathbf{0.8547 \pm 0.0221}$	$\mathbf{0.7005 \pm 0.0228}$	$\mathbf{0.7722 \pm 0.0233}$	$\mathbf{0.5701 \pm 0.0352}$
	128	$\mathbf{0.8864 \pm 0.037}$	$\mathbf{0.7618 \pm 0.0645}$	$\mathbf{0.8601 \pm 0.0148}$	$\mathbf{0.7024 \pm 0.0264}$	$\mathbf{0.7566 \pm 0.0256}$	$\mathbf{0.5419 \pm 0.0372}$
T-PINE	64	0.8148 ± 0.0318	0.6504 ± 0.0633	0.8016 ± 0.0192	0.6361 ± 0.0224	0.7033 ± 0.018	0.5204 ± 0.0185
	128	0.7989 ± 0.0297	0.6394 ± 0.0632	0.7743 ± 0.0144	0.6141 ± 0.0241	0.681 ± 0.0201	0.4822 ± 0.0166
Deepwalk	64	0.5081 ± 0.0543	0.2627 ± 0.0284	0.4786 ± 0.0206	0.2448 ± 0.0217	0.4367 ± 0.0152	0.2228 ± 0.0175
	128	0.4977 ± 0.049	0.2914 ± 0.0437	0.4674 ± 0.0207	0.2487 ± 0.0242	0.4447 ± 0.015	0.2249 ± 0.0177
Graph-AE	64	0.4591 ± 0.0306	0.1261 ± 0.0061	0.4722 ± 0.0144	0.1294 ± 0.0029	0.4767 ± 0.0079	0.1373 ± 0.0119
	128	0.4591 ± 0.0306	0.1257 ± 0.0058	0.4715 ± 0.0146	0.1281 ± 0.0027	0.4732 ± 0.0056	0.1285 ± 0.001
Graph-VAE	64	0.5261 ± 0.0322	0.2435 ± 0.0275	0.5276 ± 0.0135	0.2502 ± 0.0123	0.4985 ± 0.0165	0.2483 ± 0.0267
	128	0.542 ± 0.0446	0.2489 ± 0.0206	0.5376 ± 0.0207	0.2521 ± 0.012	0.5009 ± 0.0185	0.2434 ± 0.0258
TADW	64	0.6931 ± 0.0344	0.5368 ± 0.0562	0.6646 ± 0.0226	0.4887 ± 0.0355	0.5988 ± 0.0190	0.3889 ± 0.0250
	128	0.7511 ± 0.0404	0.6176 ± 0.0849	0.7200 ± 0.0252	0.5539 ± 0.0305	0.6287 ± 0.0213	0.4197 ± 0.0306
DGI	64	0.4705 ± 0.0378	0.147 ± 0.0199	0.4797 ± 0.0163	0.1444 ± 0.0067	0.4762 ± 0.0068	0.1336 ± 0.0036
	128	0.4727 ± 0.0374	0.1472 ± 0.0205	0.4772 ± 0.0146	0.1378 ± 0.0047	0.4746 ± 0.0069	0.1308 ± 0.0037

and some classes are skewed, macro F1 score is far more significant in this dataset. Note that Graph-AE and Graph-VAE show in general very weak classification performance and especially for the $F = 256$ in BlogCatalog they fail to produce acceptable results.

6.4.4 Link prediction

We also test the performance of the competing algorithms in the link prediction task. To do that we remove 50% of the edges for each network and then run the embedding algorithms. We form a testing set of the removed edges along with an equal number of randomly sampled non-edges. Then we compute $\boldsymbol{e}_i^T \boldsymbol{e}_j$ for each i, j edge in the testing set and rank the edges according to $\boldsymbol{e}_i^T \boldsymbol{e}_j$. Higher ranked edges are more likely to have a link. To assess the performance of the baselines we measure the area under ROC curve (AUC) and Average Precision (Avg. Prec.). The results are presented in Table 6.5 and are averaged over 5 shuffles. We observe that for Wikipedia, the proposed GAGE and the autoencoders work similarly with Graph-VAE being slightly better. In the WebKB network GAGE works the best, whereas in BlogCatalog Graph-VAE and Graph-AE outperform GAGE and the baselines. However, taking into consideration that in node classification task GAGE works markedly better, we conclude that GAGE produces more informative node embeddings.

6.4.5 Sensitivity analysis

In this subsection we examine the effect of parameter λ in the performance of the proposed GAGE embeddings for node classification and link prediction.

First, we test the effect of λ on node classification. We set the embedding dimension equal to $F = 128$ and vary λ from 1 to 0 with step equal to 0.1. We measure micro-F1 and macro-F1 scores for $90 - 10,\ 50 - 50$ and $10 - 90$ training-testing splits. Recall that high values of λ aim to preserve the network geometry associated with the connectivity information, whereas low values of λ better preserve the attribute distances. The results for Wikipedia, WebKB and BlogCatalog are presented in Figs. 6.1, 6.2 and 6.3 respectively. We observe that classification performance is consistent for $\lambda \in [0.1, 0.9]$ and the best performance is usually achieved for $\lambda \in [0.5, 0.9]$. When $\lambda = 1$ the focus is solely on the graph and classification performance is weaker compared to all other values. This stresses the importance of network attributes in node representation learning and graph node classification.

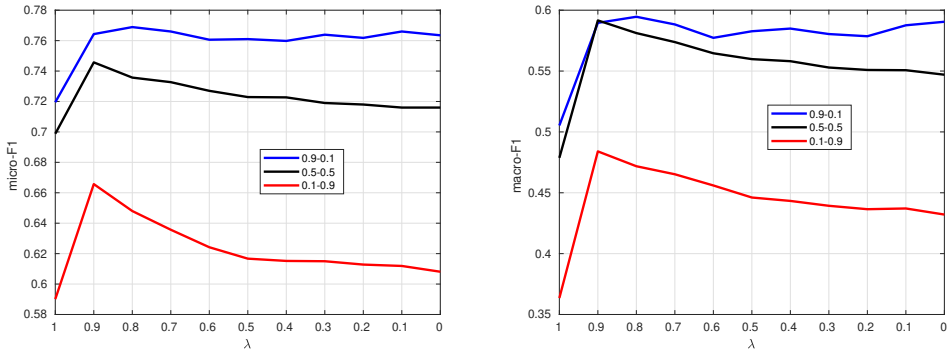
Table 6.4: Average score and standard deviation over 10 shuffles for `BlogCatalog`

Algorithm	dimension	micro (0.9)	macro (0.9)	micro (0.5)	macro (0.5)	micro (0.1)	macro (0.1)
GAGE	128	0.9233 ± 0.009	0.9208 ± 0.0095	$\mathbf{0.9191 \pm 0.0028}$	$\mathbf{0.9171 \pm 0.0027}$	$\mathbf{0.8858 \pm 0.0101}$	$\mathbf{0.8842 \pm 0.0098}$
	258	$\mathbf{0.9538 \pm 0.0082}$	$\mathbf{0.9527 \pm 0.0082}$	$\mathbf{0.9457 \pm 0.0017}$	$\mathbf{0.9447 \pm 0.0018}$	$\mathbf{0.912 \pm 0.0066}$	$\mathbf{0.9109 \pm 0.0066}$
T-PINE	128	$\mathbf{0.9281 \pm 0.0087}$	$\mathbf{0.9263 \pm 0.0093}$	0.9145 ± 0.0045	0.913 ± 0.0047	0.8577 ± 0.0055	0.8563 ± 0.0053
	256	0.9213 ± 0.0098	0.9196 ± 0.0097	0.9076 ± 0.0043	0.9061 ± 0.0044	0.8681 ± 0.0048	0.867 ± 0.0048
Deepwalk	128	0.6937 ± 0.0212	0.6802 ± 0.0218	0.681 ± 0.0056	0.673 ± 0.0059	0.6187 ± 0.0083	0.6117 ± 0.0081
	256	0.6923 ± 0.0197	0.6796 ± 0.0207	0.6823 ± 0.0051	0.6743 ± 0.0054	0.619 ± 0.0089	0.6121 ± 0.0086
Graph-AE	128	0.2521 ± 0.0128	0.179 ± 0.0077	0.2455 ± 0.0086	0.1806 ± 0.0133	0.2547 ± 0.0107	0.1454 ± 0.0154
	256	–	–	–	–	–	–
Graph-VAE	128	0.5306 ± 0.01	0.4896 ± 0.0119	0.5182 ± 0.0092	0.4754 ± 0.0128	0.467 ± 0.0149	0.4204 ± 0.021
	256	–	–	–	–	–	–
TADW	128	0.8504 ± 0.0106	0.8483 ± 0.012	0.8464 ± 0.0043	0.8442 ± 0.0044	0.8296 ± 0.0046	0.8284 ± 0.0042
	256	0.8485 ± 0.0102	0.8466 ± 0.0118	0.8446 ± 0.0041	0.8424 ± 0.0042	0.829 ± 0.0048	0.8278 ± 0.0042
DGI	128	0.6017 ± 0.0136	0.5624 ± 0.0136	0.5786 ± 0.0149	0.527 ± 0.0208	0.3693 ± 0.0583	0.2761 ± 0.0602
	256	0.6163 ± 0.0175	0.5642 ± 0.0188	0.595 ± 0.0157	0.5376 ± 0.0179	0.3236 ± 0.0729	0.2235 ± 0.0683

Table 6.5: Average score and standard deviation over 5 shuffles for link prediction

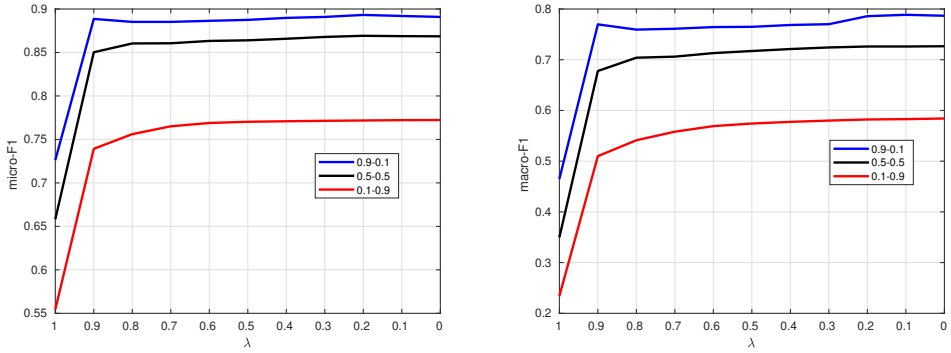
Algorithm	Dataset					
	Wikipedia		WebKB		BlogCatalog	
	AUC	Avg. Prec.	AUC	Avg. Prec.	AUC	Avg. Prec.
GAGE	0.8405 ± 0.0018	0.8819 ± 0.0017	$\mathbf{0.8604 \pm 0.0078}$	$\mathbf{0.8347 \pm 0.0112}$	0.7201 ± 0.0013	0.7589 ± 0.0077
T-PINE	0.8208 ± 0.0069	0.8767 ± 0.0044	0.7134 ± 0.0109	0.7308 ± 0.0121	0.6472 ± 0.0056	0.6638 ± 0.0034
Deepwalk	0.7965 ± 0.0056	0.8254 ± 0.0044	0.6104 ± 0.0145	0.6646 ± 0.0128	0.6645 ± 0.0049	0.6891 ± 0.0064
Graph-AE	0.8250 ± 0.0040	0.8833 ± 0.0043	0.8037 ± 0.0287	0.8335 ± 0.0225	$\mathbf{0.8231} \pm 0.0222$	0.8202 ± 0.0367
Graph-VAE	$\mathbf{0.8479 \pm 0.0073}$	$\mathbf{0.8949 \pm 0.0047}$	0.8014 ± 0.0375	0.8314 ± 0.0284	0.8218 ± 0.0100	$\mathbf{0.8248} \pm 0.0164$
TADW	0.7087 ± 0.0028	0.7722 ± 0.0032	0.7966 ± 0.0160	0.8178 ± 0.0186	0.5351 ± 0.0014	0.5317 ± 0.0008
DGI	0.8262 ± 0.0020	0.8409 ± 0.0019	0.7778 ± 0.0045	0.8179 ± 0.0032	0.7434 ± 0.0021	0.7404 ± 0.0003

Figure 6.1: Effect of λ on `Wikipedia` node classification



(a) micro-F1 score (b) macro-F1 score

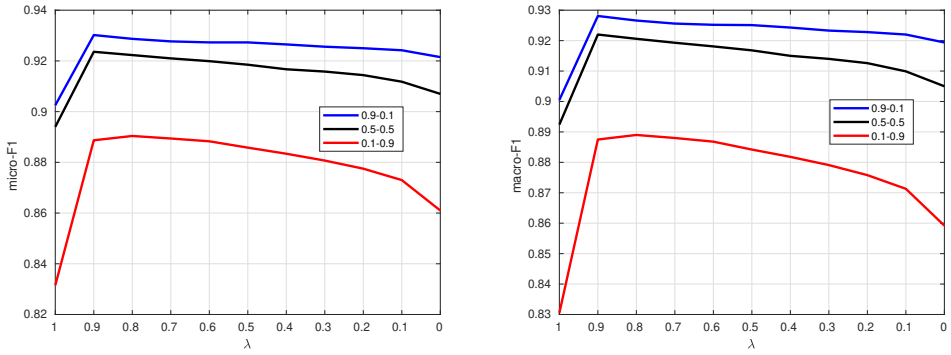
Figure 6.2: Effect of λ on `WebKB` node classification



(a) micro-F1 score (b) macro-F1 score

Next we examine the effect of parameter λ in link prediction. In this direction we vary λ from 1 to 0 with step equal to 0.1, as before, and measure the AUC and Average Precision. The embedding dimension is set to $F = 64, 128, 256$ for `WebKB`, `Wikipedia` and `BlogCatalog` respectively. The results are presented in Fig. 6.4. For `BlogCatalog` the performance is consistent across all values of λ. For `Wikipedia` and `WebKB` we observe that better link prediction is achieved when $\lambda = 1$ and the performance deteriorates as lambda decreases. This expected as potential links affect the graph geometry of the network and with $\lambda = 1$ we focus on preserving the connectivity distances between the nodes.

Figure 6.3: Effect of λ on `BlogCatalog` node classification



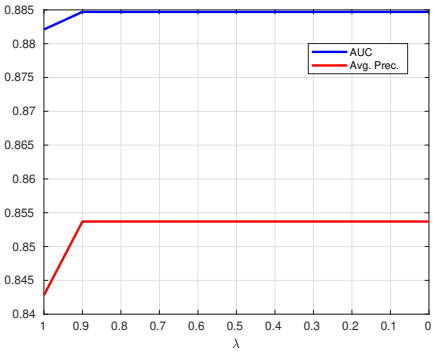
(a) micro-F1 score

(b) macro-F1 score

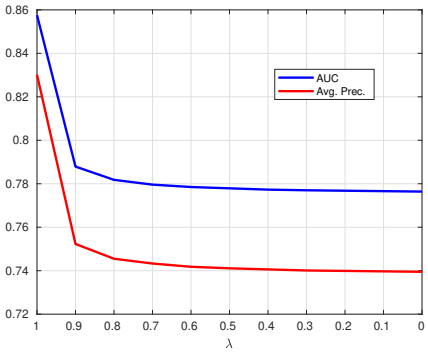
6.5 Conclusions

In this chapter we proposed `GAGE`, a novel tensor-based approach for unsupervised node embedding of attributed networks. `GAGE` leverages the favorable properties of multi dimensional scaling and canonical polyadic decomposition and provides embeddings that preserve the geometry of both network connectivity and attributes. Although the proposed approach works with distance matrices rather than the original adjacency and attributes the algorithm can still exploit the sparsity structure of the graph and the attributes and admits a scalable and lightweight implementation. Experiments with real world benchmark networks showcase the effectiveness of the proposed `GAGE` on the downstream tasks of node classification and link prediction.

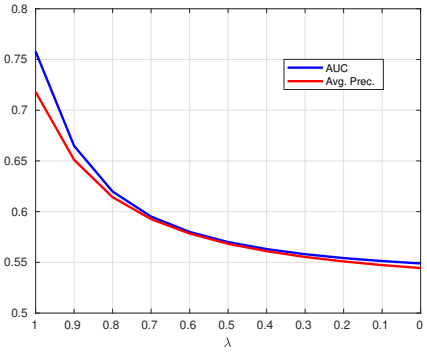
Figure 6.4: Effect of λ on link prediction



(a) Wikipedia



(b) WebKB



(c) BlogCatalog

Chapter 7

TeX-Graph: Coupled tensor-matrix knowledge-graph embedding for COVID-19 drug repurposing

How does COVID-19 relate to better-studied viral infections and biological mechanisms? Can we use existing drugs to effectively treat COVID-19 symptoms? Since the COVID-19 pandemic has disrupted our lives, there is a pressing need to answer such questions, and COVID-19 research has swiftly risen to the top of the scientific agenda, worldwide. While these questions will ultimately be answered by medical experts, data-driven methods can help to cut-down the immense search space, thus helping to accelerate progress and optimize the allocation of precious research resources. In this chapter, our goal is to derive such a method by knowledge graph embedding. We leverage tensor factorization tools to learn concise representations of entities and relations in knowledge bases and employ these representations to perform drug repurposing for COVID-19. Our proposed framework is principled, elegant, and achieves 100% improvement over the best baseline in the COVID-19 drug repurposing task using a recently developed biological KG. Part of the work presented in this Chapter is published in [87].

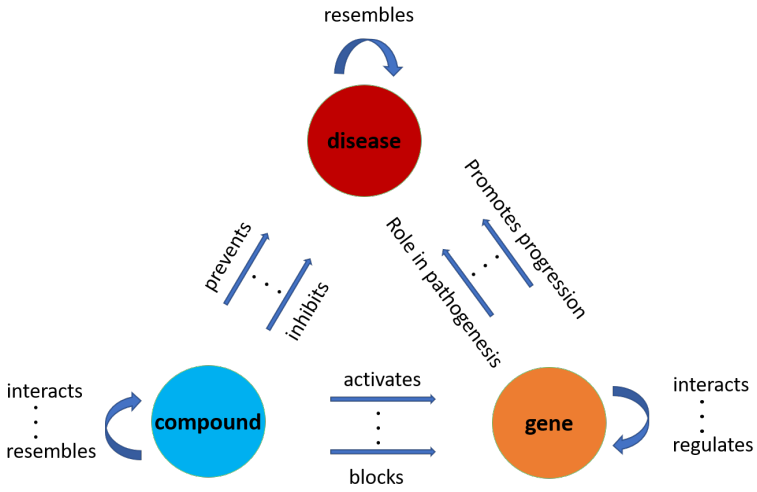


Figure 7.1: Schematic representation of biological KG.

7.1 Problem Statement

As mentioned in the introduction knowledge graphs (KGs) have attracted significant attention over the past decade due to their tremendous modeling capabilities. In particular, KGs model triplets of subject-predicate-object, denoted as (head, relation, tail) or (h, r, t). Subjects (heads) and objects (tails) are entities that are represented as graph nodes and predicates (relations) define the type of edge according to which the subject is connected to the object. A schematic representation of a KG, which models relations between genes, compounds and diseases is presented in Fig. 7.1.

In this chapter, we focus our attention on a biological KG that models relational triplets between biological entities. For example, (compound 1, interacts with, compound 2), (compound 1, activates, gene 1), (gene 1, regulates, gene 2), (compound 1, prevents, disease 1), (gene 2, is linked with, disease 2) are common triplets in numerous recently developed knowledge bases [67, 68, 74]. Modeling these types of relations as a KG enables embedding entities and relations in a Euclidean space which can further facilitate any type of processing and analysis. For instance, obtaining a low dimensional representation of compounds, diseases and the 'prevents' relation allows measuring similarity, and thus predicting and testing hypotheses regarding (compound, prevents, disease) interactions. Drug repurposing can be performed by predicting candidate compounds for new and existing target diseases. Note that the proposed framework to be introduced shortly is not limited to biological KGs – it can be applied to a wide variety of

interesting KGs.

7.1.1 Prior Art

Several methods have been proposed to learn low dimensional representations of KGs [15, 28, 53, 77, 94, 105, 119, 134, 135, 151, 156, 181]. The most popular among them adopt a single-layer perceptron or neural network approach e.g., [28, 105, 151, 156, 181]. Various tensor factorization models have also been proposed, e.g., [15, 53, 94, 119, 134]. Matrix factorization is also a tool that has been utilized for KG embedding, e.g., [77, 135].

To properly describe the most effective among them we need to define the score function $f(\cdot)$ and the loss function $\mathcal{L}(\cdot)$. Let (h_n, r_n, t_n) be an available triplet and $\boldsymbol{h}_n \in \mathbb{R}^F,\ \boldsymbol{t}_n \in \mathbb{R}^F$ and $\boldsymbol{r}_n \in \mathbb{R}^d$ be the low dimensional embeddings we aim to learn. Note that entity and relation embeddings need not be of the same dimension. The score function determines the relation model between the head (subject) and the tail (object). In simple words, high values of the score function $f(\boldsymbol{h}_n, \boldsymbol{r}_n, \boldsymbol{t}_n)$ are desirable for existing triplets (h_n, r_n, t_n) and low values of $f(\boldsymbol{h}_n, \boldsymbol{r}_n, \boldsymbol{t}_n)$ for non-existing ones.

In order to produce the entity and relational embeddings we define the following forward model for each triplet (h_n, r_n, t_n):

$$\boldsymbol{h}_n = \gamma\left(\boldsymbol{W}_e^T \boldsymbol{o}_n^h\right) \in \mathbb{R}^F, \tag{7.1a}$$

$$\boldsymbol{t}_n = \gamma\left(\boldsymbol{W}_e^T \boldsymbol{o}_n^t\right) \in \mathbb{R}^F, \tag{7.1b}$$

$$\boldsymbol{r}_n = \delta\left(\boldsymbol{W}_r^T \boldsymbol{o}_n^r\right) \in \mathbb{R}^d, \tag{7.1c}$$

where $\boldsymbol{o}_n^h \in \{0,1\}^{L_e},\ \boldsymbol{o}_n^t \in \{0,1\}^{L_e},\ \boldsymbol{o}_n^r \in \{0,1\}^{K_r}$ are one-hot input vectors corresponding to the head, tail and relation index of the triplet (h_n, r_n, t_n) respectively, with $L_e,\ K_r$ being the total number of entities (nodes) and types of relations; $\gamma(\cdot)$ and $\delta(\cdot)$ are element-wise functions and $\boldsymbol{W}_e \in \mathbb{R}^{L_e \times F},\ \boldsymbol{W}_r \in \mathbb{R}^{K_r \times d}$ are matrices that contain the model parameters to be learned.

Popular choices for $\gamma(\cdot)$ and $\delta(\cdot)$ are the identity function and hyperbolic tangent. If $\gamma(\cdot)$ or $\delta(\cdot)$ are identity functions then the rows of $\boldsymbol{W}_e$ or $\boldsymbol{W}_e$ are the learned embeddings for entities and relations respectively. For TransE, DistMult and RotatE $F = d$, whereas for TransR and

RESCAL $d \neq F$. In the TransR model $\boldsymbol{M}_r \in \mathbb{R}^{d \times F}$ is a projection matrix associated with relation r and in RESCAL $\boldsymbol{R} \in \mathbb{R}^{F \times F}$.

Table 7.1: Knowledge Graph models

Model	score function $f(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t})$
TransE [28]	$1 - \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_2$ or $1 - \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_1$
TransR [105]	$1 - \|\boldsymbol{M}_r\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{M}_r\boldsymbol{t}\|_2$
DistMult [181]	$\boldsymbol{h}^T \text{diag}(\boldsymbol{r})\boldsymbol{t}$
RESCAL [119]	$\boldsymbol{h}^T\boldsymbol{R}\boldsymbol{t}$
RotatE [156]	$1 - \|\boldsymbol{h} * \boldsymbol{r} - \boldsymbol{t}\|$

In order to learn the embeddings, state-of-the-art methods attempt to minimize the following risk:

$$\text{minimize}_{\boldsymbol{W}_e, \boldsymbol{W}_r} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(y_n - f(\boldsymbol{h}_n, \boldsymbol{r}_n, \boldsymbol{t}_n)\right) \tag{7.2}$$

where N is the number of data points (triplets or non-triplets), $y_n = 1$ if the triplet (h_n, r_n, t_n) exists, else $y_n = 0$. Typical loss functions include the logistic loss, square loss, pairwise ranking loss, margin-based ranking loss and variants of them. In order to tackle the problem in (7.2) the most popular approach is stochastic gradient descent (SGD) or batch SGD [29].

7.1.2 The 3-way model

Modeling a KG using a third order tensor has been considered in [15, 53, 94, 119, 134]. In these works, the first and second mode of the tensor is the concatenation of all the available entities, regardless of their type, whereas the third mode represents the different type of relations – i.e., each frontal slab of the third order tensor represents a certain interaction type between the entities of the KG. The methods in [15, 134] work with incomplete tensors, whereas [53, 94, 119] model each frontal slab as an adjacency matrix. To be more precise, let $\underline{\boldsymbol{Z}} \in \{0, 1\}^{L_e \times L_e \times K_r}$ be the third order tensor in [53, 94, 119]. Then $\underline{\boldsymbol{Z}}(i, j, k) = 1$ if entity i interacts with entity j through relation k and $\underline{\boldsymbol{Z}}(i, j, k) = 0$ if there is no interaction between entities i and j via the k relation.

An important observation is that although the first and second mode of tensor $\underline{\boldsymbol{Z}}$ represent the same entities, each frontal slab $\boldsymbol{Z}^k$ is not necessarily symmetric. The reason is that subject-predicate-object does not necessarily imply object-predicate-subject. The works in [53, 94]

compute the CPD of $\underline{\boldsymbol{Z}}$ (or scaled versions of $\underline{\boldsymbol{Z}}$) and produce two embeddings for each entity, one as a subject and another as an object. Although this is not always a drawback, it can result in an overparametrized model because in many applications entities usually act *either* as a subject *or* as an object, but not both. Furthermore, a single unified representation is usually preferable. In order to overcome this issue, RESCAL [119] proposed the following model for each frontal slab:

$$\boldsymbol{Z}^k = \boldsymbol{A}\boldsymbol{R}^k\boldsymbol{A}^T, \quad k = 1, \ldots, K_r, \tag{7.3}$$

where $\boldsymbol{R}^k \in \mathbb{R}^{F \times F}$ is square matrix holding the relation embeddings associated with relation k. Note that the RESCAL model is different than the traditional CPD (symmetric in mode 1 and 2) in the sense that $\boldsymbol{R}^k$ is not constrained to be diagonal. Relaxing the diagonal constraints allows matrix $\boldsymbol{R}^k$ to absorb in the relation embedding the direction in which different entities interact. On the downside, this type of relaxation forfeits the parsimony and uniqueness properties of the CPD. This is an important point, since uniqueness is a prerequisite for model interpretability when we are interested in exploratory / explanatory analysis (and not simply in making 'black box' predictions).

Another important drawback of the tree-way model is that it models unnecessary interactions. To see this, consider a KG that describes interactions between genes and diseases. Suppose that the observed interactions are of gene-gene and gene-disease type but there are no available data for disease-disease interactions. The tree-way model involves disease-disease interactions in the learning process (as non-edges), even though there are no data to justify it. As we will see in the upcoming section our proposed coupled tensor-matrix modeling addresses all the aforementioned challenges.

7.2 The `TeX-Graph` model

In this chapter we leverage coupled tensor-matrix factorization to extract low dimensional representations of entities (head, tail) as well as representations for the interactions (relation). KGs can be naturally represented by a collection of tensors and matrices, as shown in Fig. 7.2. To see this, consider the previous example of gene, compound and disease entities. Gene-compound interactions, of a certain type, can be represented by an adjacency matrix. Since there are multiple types of interactions, multiple adjacency matrices are necessary to model every interaction, resulting in a tensor $\underline{\boldsymbol{X}}_{g,c} \in \{0,1\}^{L_g \times L_c \times K_{g,c}}$, where $L_g,\ L_c$ are the number of

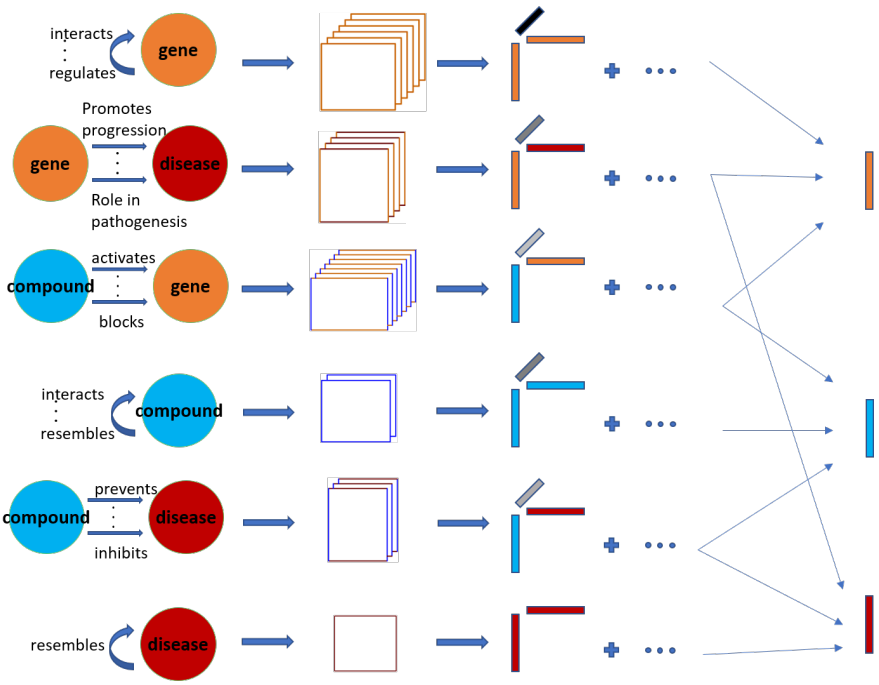


Figure 7.2: Schematic representation of `TeX-Graph` model.

genes and compounds respectively, and $K_{g,c}$ is the number of different interactions between genes and compounds. The same idea can be applied to any (entity,interaction,entity) triplet.

To facilitate the discussion let $\underline{\boldsymbol{X}}_{m,n} \in \{0,1\}^{L_m \times L_n \times K_{m,n}}$ be the tensor of interactions between entity of type-m and type-n, e.g., m codifies genes and n codifies compounds. Also let L_T be the total number of different entity types, then $m, n \in \{1, \ldots, L_T\}$. $\underline{\boldsymbol{X}}_{m,n}(i,j,k) = 1$ if the i-th entity of type-m interacts with the j-th entity of type-n via relation k and $\underline{\boldsymbol{X}}_{m,n}(i,j,k) = 0$ if there is no type-k interaction between the i-th entity of type-m and the j-th entity of type-n. The KG is represented by a collection of tensors as:

$$\begin{aligned} &\underline{\boldsymbol{X}}_{m,n} \in \{0,1\}^{L_m \times L_n \times K_{m,n}}, \ (m,n) \in \mathcal{S} \\ &\mathcal{S} = \{(m,n) : \ m \leq n, \ \exists \, (h,r,t) \text{ with } (h,t) \ \in \text{type } (m,n) \text{ or } (n,m)\}, \end{aligned} \tag{7.4}$$

where $\sum_{n=1}^{L_T} L_n = L_e$ and $\sum_{(m,n)\in\mathcal{S}} K_{m,n} = K_r$. Note that tensors $\underline{\boldsymbol{X}}_{m,n}$ and $\underline{\boldsymbol{X}}_{n,m}$ contain the same information since $\boldsymbol{X}^k_{m,n} = \boldsymbol{X}^{k^T}_{n,m}$. Therefore we only consider (m,n) tuples where $m \leq n$.

Each of the tensors in the array $\{\underline{\boldsymbol{X}}_{m,n},\ (m,n) \in \mathcal{S}\}$ admits a CPD and the overall model is cast as:

$$\underline{\boldsymbol{X}}_{m,n} = [\![\boldsymbol{A}_m, \boldsymbol{A}_n, \boldsymbol{C}_{m,n}]\!],\ (m,n) \in \mathcal{S}, \tag{7.5}$$

where $\boldsymbol{A}_n \in \mathbb{R}^{L_n \times F}, \boldsymbol{C}_{m,n} \in \mathbb{R}^{K_{m,n} \times F}$. The i-th row of $\boldsymbol{A}_n$ represents the F-dimensional embedding of the i-th type-n entity and the k-th row of $\boldsymbol{C}_{m,n}$ represents the F-dimensional embedding of the k-th type relation between type-m and type-n entities. Note that in the case where entities of type-m interact with entities of type-n via only one type of relation, $\boldsymbol{X}_{m,n} \in \{0,1\}^{L_m \times L_n}$ is a matrix and can be factored as:

$$\boldsymbol{X}_{m,n} = \boldsymbol{A}_m \text{diag}(\boldsymbol{c}_{m,n}) \boldsymbol{A}_n{}^T \tag{7.6}$$

The model in (7.5) is a coupled CPD as the factors $\boldsymbol{A}_n$ appear in multiple tensors. For instance, type-1-type-1 interactions (gene-gene), type-1-type-2 interactions (gene-compound), type-1-type-3 interactions (gene-disease), result in the factor $\boldsymbol{A}_1$ appearing in tensors $\underline{\boldsymbol{X}}_{1,1} = [\![\boldsymbol{A}_1, \boldsymbol{A}_1, \boldsymbol{C}_{1,1}]\!], \underline{\boldsymbol{X}}_{1,2} = [\![\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{C}_{1,1}]\!]$ and $\underline{\boldsymbol{X}}_{1,3} = [\![\boldsymbol{A}_1, \boldsymbol{A}_3, \boldsymbol{C}_{1,3}]\!]$.

The proposed `TeX-Graph` exhibits several favorable properties. First, the produced embeddings are unique, provided that they appear in more than one adjacency matrices.

Proposition 7.1. (Uniqueness of the embeddings) *If the coupled tensor model in* (7.5) *is indeed low-rank, F, there exist entity and relation embedding vectors in F dimensional space that generate the given knowledge base. Then the $F-$dimensional* `TeX-Graph` *embeddings for type-n entities and type-(m,n) relations are unique and permutation invariant provided that $\sum_{m \in \mathcal{S}_n^+} K_{m,n} + \sum_{p \in \mathcal{S}_n^-} K_{n,p} > 1$ and $K_{m,n} > 1$ respectively, where $\mathcal{S}_n^+,\ \mathcal{S}_n^-$ are defined in* (7.9).

Proof sketch: In order to prove Proposition 7.1 we choose the subtensor $\underline{\boldsymbol{X}}_{m,n}$ such that $\boldsymbol{A}_m$ or $\boldsymbol{A}_n$ are the most frequent factors in the coupled model— suppose this is $\boldsymbol{A}_n$. Using the uniqueness conditions of Theorem 2.4 we can establish identifiability of all $\boldsymbol{A}_n,\ \boldsymbol{A}_m,\ \boldsymbol{A}_p,\ \boldsymbol{C}_{m,n},\ C_{n,p}$ such that $m \in \mathcal{S}_n^+,\ p \in \mathcal{S}_n^-$. For the remaining factors can be identified as solutions to a system of linear equations.

In the case where $K_{m,n} = 1$ and type-m entities appear in multiple tensors but type-n entities only in one, the `TeX-Graph` model identifies $\boldsymbol{A}_m$ and $\boldsymbol{A}_n \text{diag}(\boldsymbol{c}_{m,n})$, since there is rotational freedom between $\boldsymbol{A}_n$ and $\boldsymbol{c}_{m,n}$.

Another important property of the proposed TeX-Graph is that it avoids modeling of spurious 'cross-product' relations that can never be observed. The coupled tensor-matrix model allows for a concise KG representation that eliminates such spurious relations from the start, contrary to the three-way model. To see this, consider the previous example of gene-disease KG that observes relational triplets between gene-gene and gene-disease type but not for disease-disease type. The proposed TeX-Graph does not model disease-disease interactions, whereas the three-way model treats them as non-edges.

It is worth noticing that TeX-Graph makes the implicit assumption that $\underline{\boldsymbol{X}}_{n,n}$ are symmetric in the first and second mode. This is not always the case, since interactions between some entity types might be directed. To overcome this issue we assume that (h,r,t) implies (t,r,h) for (h,t) of the same type. Although this assumption ignores the direction in this type of interactions, it results in a more parsimonious model for the entity embeddings.

7.2.1 Algorithmic framework

In order to learn the F-dimensional embeddings of all entities and relations we formulate the KG embedding problem as:

$$\text{minimize}_{\{\boldsymbol{A}_m\},\{\boldsymbol{C}_{m,n}\}} \sum_{(m,n)\in\mathcal{S}} \left\| \underline{\boldsymbol{X}}_{m,n} - [\![\boldsymbol{A}_m, \boldsymbol{A}_n, \boldsymbol{C}_{m,n}]\!] \right\|_F^2, \tag{7.7}$$

The problem in (7.7) is non-convex and NP-hard in general. In order to tackle it we propose to fix all variables but one and update the remaining variable. This procedure is repeated in an alternating fashion. The update for $\boldsymbol{A}_n$ is a system of linear equations and takes the form:

$$\sum_{m\in\mathcal{S}_n^+} \left(\boldsymbol{C}_{m,n} \odot \boldsymbol{A}_m\right)^T \boldsymbol{X}_{m,n}^{(2)} + \sum_{p\in\mathcal{S}_n^-} \left(\boldsymbol{C}_{n,p} \odot \boldsymbol{A}_p\right)^T \boldsymbol{X}_{n,p}^{(1)} = \tag{7.8}$$

$$\left(\sum_{m\in\mathcal{S}_n^+} \left(\boldsymbol{C}_{m,n}^T \boldsymbol{C}_{m,n} * \boldsymbol{A}_m^T \boldsymbol{A}_m \right) + \sum_{p\in\mathcal{S}_n^-} \left(\boldsymbol{C}_{n,p}^T \boldsymbol{C}_{n,p} * \boldsymbol{A}_p^T \boldsymbol{A}_p \right) \right) \boldsymbol{A}_n^T,$$

where

$$\mathcal{S}_n^+ = \{m : (m,n) \in \mathcal{S}\},\ \mathcal{S}_n^- = \{p : (n,p) \in \mathcal{S}\}. \tag{7.9}$$

Algorithm 7.1 `TeX-Graph`

Input: $\{(h_n, r_n, t_n)\}_{n=1}^{N},\ \{\boldsymbol{A}_m\},\ \{\boldsymbol{C}_{m,n}\}$.

Output: $\{\boldsymbol{A}_n\}_{n=1}^{L_e},\ \{\boldsymbol{C}_{m,n}\}_{(m,n)\in\mathcal{S}}$.

Create $\{\underline{\boldsymbol{X}}_{m,n}\}_{(m,n)\in\mathcal{S}}$ from $\{(h_n, r_n, t_n)\}_{n=1}^{N}$;

repeat

  for $n \in \{1, \ldots, L_E\}$ **do**

    $\boldsymbol{A}_n \leftarrow$ solve (7.8);

  end for

  for $(m, n) \in \mathcal{S}$ **do**

    $\boldsymbol{C}_{(m,n)} \leftarrow$ solve (7.10);

  end for

until criterion is met.

Algorithm 7.2 `TeX-Graph-initialization`

Input: $\{(h_n, r_n, t_n)\}_{n=1}^{N}$.

Output: $\{\boldsymbol{A}_n\}_{n=1}^{L_e},\ \{\boldsymbol{C}_{m,n}\}_{(m,n)\in\mathcal{S}}$.

Create tensor $\underline{\boldsymbol{Z}}$ from $\{(h_n, r_n, t_n)\}_{n=1}^{N}$ as explained in section 7.1.2;

Form $\underline{\boldsymbol{Y}}$ as: $\underline{\boldsymbol{Y}}(i,j,k) = \min\{1, \underline{\boldsymbol{Z}}(i,j,k) + \underline{\boldsymbol{Z}}(j,i,k)\}$;

Solve $\underline{\boldsymbol{Y}} = [\![\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{C}]\!]$ via sparse EVD;

Form $\{\boldsymbol{A}_n\}_{n=1}^{L_e},\ \{\boldsymbol{C}_{m,n}\}_{(m,n)\in\mathcal{S}}$ from $\boldsymbol{A},\ \boldsymbol{C}$.

The update for $\boldsymbol{C}_{m,n}$ is the solution to the following system of linear equations:

$$(\boldsymbol{A}_n \odot \boldsymbol{A}_m)^T \boldsymbol{X}_{m,n}^{(3)} = \left(\boldsymbol{A}_n^T \boldsymbol{A}_n * \boldsymbol{A}_m^T \boldsymbol{A}_m\right) \boldsymbol{C}_{m,n}^T \tag{7.10}$$

The derivations for these updates as well as implementation details are presented in Appendix E.

The proposed `TeX-Graph` is presented in Algorithm 7.1. `TeX-Graph` is an iterative algorithm that tackles a non-convex problem and NP-hard in general. As a result different initial points might produce different results. Although we have observed that random initialization is sufficient most of the times we propose an alternative initialization procedure that yields consistent and reproducible results. To be more specific we form a symmetric version of tensor $\underline{\boldsymbol{Z}}$ as:

$$\underline{\boldsymbol{Y}}(i,j,k) = \min\{1, \underline{\boldsymbol{Z}}(i,j,k) + \underline{\boldsymbol{Z}}(j,i,k)\} \tag{7.11}$$

Then we compute the semi-symmetric CPD of $\underline{\boldsymbol{Y}} = [\![\boldsymbol{A}, \boldsymbol{A}, \boldsymbol{C}]\!]$ using sparse eigenvalue decomposition (EVD) [136]. The proposed initialization procedure is presented in Algorithm 7.2.

7.2.2 Computational complexity analysis

In terms of memory requirements and computational complexity, the main bottleneck of `TeX-Graph` lies in instantiating and computing the matricized tensor times Khatri-Rao product (MTTKRP) in the left hand side (LHS) of (7.8) and (7.10). The number of flops needed to compute the LHS of (7.8) and (7.10) is $\mathcal{O}\left(F \cdot \text{nnz}\left(\sum_{m\in\mathcal{S}_n^+} \underline{\boldsymbol{X}}_{m,n} + \sum_{p\in\mathcal{S}_n^-} \underline{\boldsymbol{X}}_{n,p}\right)\right)$ and $\mathcal{O}\left(F \cdot \text{nnz}\left(\underline{\boldsymbol{X}}_{m,n}\right)\right)$ respectively. For small values of F which is usually the case in practice the complexity is linear in the number of triplets participating in each update. Furthermore the Khatri-Rao products in the (LHS) of (7.8) and (7.10) are not being instantiated as shown in Appendix E.

7.3 Drug Repurposing for COVID-19

In this section we apply `TeKGraph` to a recently developed KG [74] in order to perform drug repurposing for COVID-19 disease. All algorithms were implemented in Matlab or Python, and executed on a Linux server comprising 32 cores at 2GHz and 128GB RAM.

7.3.1 Data

The dataset used in the experiments is the Drug Repurposing Knowledge Graph (DRKG)[1] [74]. It codifies triplets of biological interactions between 97,238 different entities of 13 types, namely, genes, compounds, diseases, anatomy, tax, biological process, cellular component, pathway, molecular function, anatomical therapeutic chemical (Atc), side effect, pharmacological class, and symptom. The total number of triplets is 5,874,258 and there are 107 different types of interactions. The KG is organised in 6 adjacency tensors and 11 adjacency matrices. Detailed description of the dataset and the modeling can be found in Table 7.2. Each row denotes a different adjacency tensor or matrix and $\#$ type-m entities, $\#$ type-m entities, $\#$ relation types

[1] github.com/gnn4dr/DRKG

correspond to the dimension of mode 1, mode 2, and mode 3 respectively. The last column (sparsity) denotes the sparsity of each tensor, i.e., $\frac{\text{nnz}(\underline{\boldsymbol{X}}_{m,n})}{L_m L_n K_{m,n}}$.

7.3.2 Procedure

Drug repurposing refers to the task of discovering existing drugs that can effectively manage certain diseases– COVID-19 in our study. DRKG codifies relational triplets of (compound,treats,disease) and (compound,inhibits,disease). Therefore drug repurposing in the context of DRKG boils down to predicting new 'treats' and 'inhibits' edges (links) between compounds and diseases of interest.

We follow the evaluation procedure proposed in [74]. In the training phase we learn low dimensional representations for the entities and relations, using all the edges in DRKG. In the testing phase, we assign a score to (compound,treats,disease) and (compound,inhibits,disease) triplets according to the scoring function used for training. For the proposed `TeX-Graph`, the scores assigned to the triplet (hyper-edge) (compound i,treats,disease j) and (compound i,inhibits,disease j) are:

$$\text{score}_{i,j,2} = \boldsymbol{A}_2(i,:)\text{diag}\left(\boldsymbol{C}_{2,3}\left(2,:\right)\right)\boldsymbol{A}_2(j,:)^T,$$

$$\text{score}_{i,j,9} = \boldsymbol{A}_2(i,:)\text{diag}\left(\boldsymbol{C}_{2,3}\left(9,:\right)\right)\boldsymbol{A}_2(j,:)^T,$$

since 'treats' and 'inhibits' relations correspond to the second and ninth frontal slab of $\underline{\boldsymbol{X}}_{2,3}$, respectively. The testing set consists of 34 corona-virus related diseases, including SARS, MERS and SARS-COV2 and 8,103 FDA-approved drugs in Drugbank. Drugs with molecule weight less than 250 daltons are excluded from testing. Ribavirin was also excluded from the testing set, since there exist a 'treat' edge in the training set between Ribavirin and a target disease. In order to evaluate the performance of the proposed `TeX-Graph` and the alternatives we retrieve the top-100 ranked drugs that appear in the highest testing scoring (hyper-)edges. These are the proposed candidate drugs for COVID-19. Then we assess how many of the 32 clinical trial drugs [2] (Ribavirin is excluded) appear in the proposed candidate top-100 drugs.

[2] www.covid19-trials.com

Table 7.2: Coupled tensor-matrix DRKG modeling.

entity type-m	entity type-n	# type-m entities 1	# type-n entities 2	# relation types	tensor	sparsity
Gene	Gene	39,220	39,220	32	$\underline{\boldsymbol{X}}_{1,1} = [\![\boldsymbol{A}_1, \boldsymbol{A}_1, \boldsymbol{C}_{1,1}]\!]$	$6.12\ 10^{-5}$
	Compound	39,220	24,313	34	$\underline{\boldsymbol{X}}_{1,2} = [\![\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{C}_{1,2}]\!]$	$6.50\ 10^{-6}$
	Disease	39,220	5,103	15	$\underline{\boldsymbol{X}}_{1,3} = [\![\boldsymbol{A}_1, \boldsymbol{A}_3, \boldsymbol{C}_{1,3}]\!]$	$4.13\ 10^{-5}$
	Anatomy	39,220	400	3	$\underline{\boldsymbol{X}}_{1,4} = [\![\boldsymbol{A}_1, \boldsymbol{A}_4, \boldsymbol{C}_{1,4}]\!]$	0.0154
	Tax	39,220	215	1	$\boldsymbol{X}_{1,5} = \boldsymbol{A}_1 \text{diag}(\boldsymbol{c}_{1,5}) \boldsymbol{A}_5^T$	0.0017
	Biological Process	39,220	11,381	1	$\boldsymbol{X}_{1,6} = \boldsymbol{A}_1 \text{diag}(\boldsymbol{c}_{1,6}) \boldsymbol{A}_6^T$	0.0013
	Cellular Component	39,220	1,391	1	$\boldsymbol{X}_{1,7} = \boldsymbol{A}_1 \text{diag}(\boldsymbol{c}_{1,7}) \boldsymbol{A}_7^T$	0.0013
	Pathway	39,220	1,822	1	$\boldsymbol{X}_{1,8} = \boldsymbol{A}_1 \text{diag}(\boldsymbol{c}_{1,8}) \boldsymbol{A}_8^T$	0.0012
	Molecular Function	39,220	2,884	1	$\boldsymbol{X}_{1,9} = \boldsymbol{A}_1 \text{diag}(\boldsymbol{c}_{1,9}) \boldsymbol{A}_9^T$	8.610^{-4}
Compound	Compound	24,313	24,313	2	$\underline{\boldsymbol{X}}_{2,2} = [\![\boldsymbol{A}_2, \boldsymbol{A}_2, \boldsymbol{C}_{2,2}]\!]$	0.0023
	Disease	24,313	5,103	10	$\underline{\boldsymbol{X}}_{2,3} = [\![\boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{C}_{2,3}]\!]$	$6.76\ 10^{-5}$
	Atc	24,313	4,048	1	$\boldsymbol{X}_{2,10} = \boldsymbol{A}_2 \text{diag}(\boldsymbol{c}_{2,10}) \boldsymbol{A}_{10}^T$	$1.6\ 10^{-4}$
	Side Effect	24,313	5,701	1	$\boldsymbol{X}_{2,11} = \boldsymbol{A}_2 \text{diag}(\boldsymbol{c}_{2,11}) \boldsymbol{A}_{11}^T$	0.0010
	Pharmacological Class	24,313	345	1	$\boldsymbol{X}_{2,12} = \boldsymbol{A}_2 \text{diag}(\boldsymbol{c}_{2,12}) \boldsymbol{A}_{12}^T$	$1.22\ 10^{-4}$
Disease	Disease	5,103	5,103	1	$\boldsymbol{X}_{3,3} = \boldsymbol{A}_3 \text{diag}(\boldsymbol{c}_{3,3}) \boldsymbol{A}_3^T$	$4.17\ 10^{-5}$
	Anatomy	5,103	400	1	$\boldsymbol{X}_{3,4} = \boldsymbol{A}_3 \text{diag}(\boldsymbol{c}_{3,4}) \boldsymbol{A}_4^T$	0.0018
	Symptom	5,103	415	1	$\boldsymbol{X}_{3,13} = \boldsymbol{A}_3 \text{diag}(\boldsymbol{c}_{3,13}) \boldsymbol{A}_{13}^T$	0.0016

7.3.3 Methods

The methods used in the experiments are:

- **TeX-Graph**. The proposed `TeKGraph` algorithm initialized with Algorithm 7.2. The embedding dimension is set to $F = 50$ and the algorithm runs for 10 iterations.
- **TransE-DRKG** [28, 74]. `TransE` learns low dimensional KG embeddings using the score function shown in Table 7.1. For the the task of drug repurposing we use the specifications proposed in [74]. The l_2 norm is chosen in the score function and training is performed using the deep graph library for knowledge graphs [191]. To evaluate the performance of TransE-DRKG on the drug repurposing task we used the $400-$dimensional pretrained embeddings in [74], with which the drug repurposing results were better than the stand-alone code without pretraining.
- **3-way KG embeddings (3-way KGE)**. We add as a baseline the embeddings produced by computing the CPD of tensor $\underline{\boldsymbol{Y}}$ in (7.11). Recall that we use an algebraic CPD of $\underline{\boldsymbol{Y}}$ to initialize `TeX-Graph`. In `3-way KGE` we initialize using the same procedure and also run 10 alternating least-squares iterations to compute the CPD of $\underline{\boldsymbol{Y}}$. `3-way KGE` is tested with $F = 50$.

7.3.4 Results

Table 7.3 shows the clinical trial drugs that appear in the top-100 recommendations along with their [rank-order]. The proposed approach retrieves 10 clinical trial drugs in the top-100 positions, and 7 in the top-50. Compared to `TransE-DRKG` that was the first proposed algorithm to perform drug-repurposing for COVID-19, `TeX-Graph` achieves 75% and 100% improvement in precision in the top-50 and top-100 respectively.

It is worth emphasizing that the proposed `Tex-Graph` retrieves approximately $1/3$ of the COVID-19 clinical trial drugs, in the top-100, among a testing set of $8,103$ drugs. This result is pretty remarkable and can essentially help cutting down the immense search space of medical research. For instance, consider the case of Dexamethasone, which is retrieved by `Tex-Graph` in the top ranked position (it admitted the highest score among all $8,103$ drugs). At the onset of the pandemic, the initial guidance for Dexamethasone and other corticosteroids was indecisive. Guidelines from different sources issued either a weak recommendation to use Dexamethasone

(with an asterisk that further evidence was required) or a weak recommendation against corticosteroids and Dexamethasone [131]. However, recent results indicate that treatment with Dexamethasone reduces mortality in patients with COVID-19 [69]. The results of `Tex-Graph` coalign with the latest evidence and rank Dexamethasone as the top recommended drug. This suggests that our proposed data-driven approach could have essentially contributed in overturning the initial hesitancy to administrate Dexamethasone as a first line treatment.

Table 7.3: Proposed candidate drugs for COVID-19

`TeX-Graph`	`TransE-DRKG`	`3-way KGE`
F=50	F=400	F=50
Dexamethasone [1]	Dexamethasone [4]	Oseltamivir [89]
Methylprednisolone [6]	Colchine [8]	
Azithromycin [13]	Methylprednisolone [16]	
Thalidomide [18]	Oseltamivir [49]	
Losartan [41]	Deferoxamine [87]	
Hydroxychloroquine [47]		
Colchine [48]		
Oseltamivir[60]		
Chloroquine[68]		
Deferoxamine [88]		

7.4 Conclusion

In this chapter we proposed a novel coupled tensor-matrix framework for knowledge graph embedding. The proposed model is principled and enjoys several favorable properties, including parsimony and uniqueness. The developed algorithmic framework admits lightweight updates and can handle very large graphs. Finally the proposed `TeX-Graph` showed very promising results in a timely application to drug repurposing, a task of paramount importance in the fight against COVID-19.

Chapter 8

Thesis Summary and Future Directions

The present thesis proposed elegant and effective frameworks to a series of machine learning and signal processing tasks. The frameworks are supported by rigorous theoretical analysis, detailed algorithmic development and thorough experimental examination.

The task of hyperspectral super-resolution (HSR) was studied in Chapter 3. A coupled CPD model was employed to perform the fusion task. In particular, learning the factors corresponding to the spatial dimensions from the multispectal image, and the factor corresponding to the spectral dimension from the hyperspectral image, allows to construct a super-resolution image that admits a high spatial and spectral resolution. An alternative that exploits the low rank matrix structure in the spectral dimension was also proposed. The two proposed methods are the first that can provably guarantee identifiability of the super-resolution image under practical conditions. Furthermore, they do not require knowledge of the spatial degradation operator, which is unknown in practice. Their performance was evaluated using real hyperspectral images. The results suggest that the proposed methods markedly outperform the baselines.

Tensor completion from regular samples was the topic of Chapter 4. Different tensor sampling models were studied along with conditions under which tensor recovery is guaranteed. The task of fMRI scan acquisition was cast as a tensor completion problem and an algorithmic framework was designed to tackle it. The tensor sampling schemes were tested with synthetic data and real raw fMRI scans, that manifested the effectiveness of the proposed framework.

Chapter 5 introduced two new regular sampling schemes. Theoretical analysis showed that computing the CPD of a regularly sampled tensor yields the same solution to the CPD of the full tensor. Two algorithms tailored to these schemes were designed to perform the CPD computation for very large tensors. Experiments with synthetic and real data showed that the proposed algorithms can compute the CPD with significant speed-up and same accuracy compared to the algorithms the operate on the full tensor.

Chapter 6 built upon a framework that learns low dimensional representations of nodes in attributed graphs. The proposed embeddings are geometry preserving, in the sense that they can reproduce the distances defined by the connectivity and attribute information of the network. A lightweight algorithm was developed that can work with very large networks. The proposed embeddings demonstrate remarkable performance in the downstream tasks of node classification and link prediction.

Finally, Chapter 7 proposed a novel tensor-matrix framework to learn low dimensional representations of entities and relations present in knowledge graphs. The proposed embedding framework was developed to perform drug repurposing in the fight against COVID-19. Compared to the baseline, the proposed tensor-matrix approach was able to retrieve twice as many clinical trial drugs in the top-100.

Based on this dissertation the author is planning to work in the following directions:

- Joint learning of tensor and matrix models: In Chapter 3 we introduced a hybrid model that was able to take advantage of the low rank tensor and matrix structure present in hyperspectral and multispectral images. In the proposed approach the low rank matrix subspace is learned independently of the low rank tensor factors and vise versa. The author plans to design algorithms to jointly learn the matrix subspace along with the tensor factors that could potentially yield better super-resolution results. Furthermore, the issue of selecting image blocks in the `SCUBA` algorithm was not discussed. The author intends to explore optimal ways to select blocks, such that the number of endmembers in each block remains less than or equal to the spectral dimension of the multispectral image.

- Optimal systematic sampling of tensor signals: In Chapters 4 and 5 we studied how to regularly sample tensors. In Chapter 4 we focused on schemes and conditions that guarantee reconstruction of the tensor, whereas in 5 the focus was on designing sampling patterns that allow fast CPD computations. In this direction, we intend to conduct research

on more effective systematic schemes. We have noticed that optimal sampling, from an identifiability viewpoint, is the one that leads the maximum Kruskal rank on the factors. In order to take advantage of this fact and design an efficient algorithm we need to relate the Kruskal rank of the factors to the original tensor. The author plans to work on this direction.

- Canonical correlation embeddings for multidimensional graphs: Identifying the relation between tensors and attributed networks or knowledge graphs was the blueprint to the work presented in Chapters 6 and 7. The next step involves exploring different models. In particular we are planning to apply canonical correlation analysis on the aforementioned graphs to extract representations that hold the common information between different views of the graph. In the case of attributed networks the views involve the connectivity and attribute distance matrices, whereas in knowledge graphs different views correspond to different interaction types. Canonical correlation analysis has proven to be a very effective tool in natural language processing [82, 154] and communication [73] tasks and therefore we expect it to be a good alternative to our proposed tensor methods.

References

[1] "Landsat satellite sensor," https://landsat.gsfc.nasa.gov, accessed: 2018-04-04.

[2] "Quickbird satellite sensor," http://www.satimagingcorp.com/satellite-sensors/quickbird, accessed: 2018-04-04.

[3] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

[4] A. Ahmed, N. Shervashidze, S. Narayanamurthy, V. Josifovski, and A. J. Smola, "Distributed large-scale natural graph factorization," in *Proceedings of the 22nd international conference on World Wide Web*, 2013, pp. 37–48.

[5] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, and M. Selva, "An mtf-based spectral distortion minimizing model for pan-sharpening of very high resolution multispectral images of urban areas," in *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on*. IEEE, 2003, pp. 90–94.

[6] B. Aiazzi, L. Alparone, S. Baronti, A. Garzelli, M. Selva, and C. Chen, "25 years of pansharpening: a critical review and new developments," *Signal and Image Processing for Remote Sensing*, pp. 533–548, 2011.

[7] B. Aiazzi, S. Baronti, and M. Selva, "Improving component substitution pansharpening through multivariate regression of ms + pan data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3230–3239, 2007.

[8] M. Akçakaya, S. Moeller, S. Weingärtner, and K. Uğurbil, "Scan-specific robust artificial-neural-networks for k-space interpolation (RAKI) reconstruction: Database-free deep learning for fast imaging," *Magnetic resonance in medicine*, 2018.

[9] S. A. Al-Sayouri, E. Gujral, D. Koutra, E. E. Papalexakis, and S. S. Lam, "t-pne: tensor-based predictable node embeddings," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 491–494.

[10] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2773–2832, 2014.

[11] M. Ashraphijuo and X. Wang, "Fundamental conditions for low-cp-rank tensor completion," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 2116–2145, 2017.

[12] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[13] B. W. Bader, R. A. Harshman, and T. G. Kolda, "Temporal analysis of semantic graphs using asalsan," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 33–42.

[14] B. W. Bader and T. G. Kolda, "Efficient matlab computations with sparse and factored tensors," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, 2007.

[15] I. Balazevic, C. Allen, and T. Hospedales, "Tucker: Tensor factorization for knowledge graph completion," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 5188–5197.

[16] D. Banco, S. Aeron, and W. S. Hoge, "Sampling and recovery of mri data using low rank tensor models," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2016, pp. 448–452.

[17] A.-L. Barabási *et al.*, *Network science*. Cambridge university press, 2016.

[18] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[19] J. A. Barsi, B. L. Markham, and J. A. Pedelty, "The operational land imager: spectral response and spectral uniformity," in *Earth Observing Systems XVI*, vol. 8153. International Society for Optics and Photonics, 2011, p. 81530G.

[20] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation ax+ xb= c [f4]," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[21] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical programming computation*, vol. 3, no. 3, p. 165, 2011.

[22] D. Berberidis and G. B. Giannakis, "Node embedding with adaptive similarities for scalable learning over graphs," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[23] M. Berlingerio, M. Coscia, F. Giannotti, A. Monreale, and D. Pedreschi, "Foundations of multidimensional network analysis," in *2011 international conference on advances in social networks analysis and mining*. IEEE, 2011, pp. 485–489.

[24] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[25] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[26] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, April 2012.

[27] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1247–1250.

[28] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," in *Advances in neural information processing systems*, 2013, pp. 2787–2795.

[29] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[30] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[31] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[32] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[33] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM international on conference on information and knowledge management*, 2015, pp. 891–900.

[34] ——, "Deep neural networks for learning graph representations." in *AAAI*, vol. 16, 2016, pp. 1145–1152.

[35] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[36] W. CARPER, T. LILLESAND, and R. KIEFER, "The use of intensity-hue-saturation transformations for merging spot panchromatic and multispectral image data," *Photogrammetric Engineering and remote sensing*, vol. 56, no. 4, pp. 459–467, 1990.

[37] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[38] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4177–4193, 2011.

[39] C. Chatzichristos, E. Kofidis, M. Morante, and S. Theodoridis, "Blind fmri source unmixing via higher-order tensor decompositions," *Journal of Neuroscience Methods*, 2018.

[40] L. Chiantini and G. Ottaviani, "On generic identifiability of 3-tensors of small rank," *SIAM Journal on Matrix Analysis and Applications*, vol. 33, no. 3, pp. 1018–1037, 2012.

[41] M. Chiew, N. N. Graedel, and K. L. Miller, "Recovering task fmri signals from highly under-sampled data with low-rank and temporal subspace constraints," *NeuroImage*, vol. 174, pp. 97–110, 2018.

[42] M. Chiew, S. M. Smith, P. J. Koopmans, N. N. Graedel, T. Blumensath, and K. L. Miller, "k-t faster: acceleration of functional mri data acquisition using low rank constraints," *Magnetic resonance in medicine*, vol. 74, no. 2, pp. 353–364, 2015.

[43] A. G. Christodoulou, G. Redler, B. Clifford, Z.-P. Liang, H. J. Halpern, and B. Epel, "Fast dynamic electron paramagnetic resonance (epr) oxygen imaging using low-rank tensors," *Journal of Magnetic Resonance*, vol. 270, pp. 176–182, 2016.

[44] M. A. Cox and T. F. Cox, "Multidimensional scaling," in *Handbook of data visualization*. Springer, 2008, pp. 315–347.

[45] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[46] S. Delvaux and M. Van Barel, "Rank-deficient submatrices of fourier matrices," *Linear Algebra and its Applications*, vol. 429, no. 7, pp. 1587–1605, 2008.

[47] I. Domanov and L. De Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensors — part ii: Uniqueness of the overall decomposition," *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, vol. 34, no. 3, pp. 876–903, 2013.

[48] I. Domanov and L. D. Lathauwer, "Canonical polyadic decomposition of third-order tensors: reduction to generalized eigenvalue decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 2, pp. 636–660, 2014.

[49] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[50] D. Easley, J. Kleinberg *et al.*, *Networks, crowds, and markets*. Cambridge university press Cambridge, 2010, vol. 8.

[51] R. C. Farias, J. E. Cohen, and P. Comon, "Exploring multimodal data fusion through joint decompositions with flexible couplings," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4830–4844, 2016.

[52] D. A. Feinberg and E. Yacoub, "The rapid development of high speed, resolution and precision in fmri," *Neuroimage*, vol. 62, no. 2, pp. 720–725, 2012.

[53] T. Franz, A. Schultz, S. Sizov, and S. Staab, "Triplerank: Ranking semantic web data by tensor decomposition," in *International semantic web conference*. Springer, 2009, pp. 213–228.

[54] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *arXiv preprint arXiv:1803.01257*, 2018.

[55] X. Fu, N. D. Sidiropoulos, J. H. Tranter, and W.-K. Ma, "A factor analysis framework for power spectra separation and multiple emitter localization," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6581–6594, 2015.

[56] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.

[57] L. Getoor, "Link-based classification," in *Advanced methods for knowledge discovery from complex data*. Springer, 2005, pp. 189–207.

[58] G. Golub, S. Nash, and C. Van Loan, "A hessenberg-schur method for the problem ax+ xb= c," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, 1979.

[59] G. Golub and C. Van Loan, "Matrix computations 4th edition the johns hopkins university press," *Baltimore, MD*, 2013.

[60] R. B. Gomez, A. Jazaeri, and M. Kafatos, "Wavelet-based hyperspectral and multispectral image fusion," in *Geo-Spatial Image and Data Exploitation II*, vol. 4383. International Society for Optics and Photonics, 2001, pp. 36–43.

[61] M. A. Griswold, P. M. Jakob, R. M. Heidemann, M. Nittka, V. Jellus, J. Wang, B. Kiefer, and A. Haase, "Generalized autocalibrating partially parallel acquisitions (GRAPPA)," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 47, no. 6, pp. 1202–1210, 2002.

[62] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[63] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[64] R. A. Harshman, M. E. Lundy *et al.*, "Parafac: Parallel factor analysis," *Computational Statistics and Data Analysis*, vol. 18, no. 1, pp. 39–72, 1994.

[65] J. Haupt, L. Applebaum, and R. Nowak, "On the restricted isometry of deterministically subsampled fourier matrices," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, March 2010, pp. 1–6.

[66] J. He, Q. Liu, A. G. Christodoulou, C. Ma, F. Lam, and Z.-P. Liang, "Accelerated high-dimensional mr imaging with sparse sampling using low-rank tensors," *IEEE transactions on medical imaging*, vol. 35, no. 9, pp. 2119–2129, 2016.

[67] D. S. Himmelstein and S. E. Baranzini, "Heterogeneous network edge prediction: a data integration approach to prioritize disease-associated genes," *PLoS computational biology*, vol. 11, no. 7, 2015.

[68] D. S. Himmelstein, A. Lizee, C. Hessler, L. Brueggeman, S. L. Chen, D. Hadley, A. Green, P. Khankhanian, and S. E. Baranzini, "Systematic integration of biomedical knowledge prioritizes drugs for repurposing," *Elife*, vol. 6, p. e26726, 2017.

[69] P. Horby, W. S. Lim, J. R. Emberson, M. Mafham, J. L. Bell, L. Linsell, N. Staplin, C. Brightling, A. Ustianowski, E. Elmahi *et al.*, "Dexamethasone in hospitalized patients with covid-19-preliminary report." *The New England journal of medicine*, 2020.

[70] V. Hore, A. Viñuela, A. Buil, J. Knight, M. I. McCarthy, K. Small, and J. Marchini, "Tensor decomposition for multiple-tissue gene expression experiments," *Nature genetics*, vol. 48, no. 9, p. 1094, 2016.

[71] B. Huang, C. Mu, D. Goldfarb, and J. Wright, "Provable low-rank tensor recovery," *Optimization-Online*, vol. 4252, p. 2, 2014.

[72] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 731–739.

[73] M. S. Ibrahim and N. D. Sidiropoulos, "Cell-edge interferometry: Reliable detection of unknown cell-edge users via canonical correlation analysis," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[74] V. N. Ioannidis, X. Song, S. Manchanda, M. Li, X. Pan, D. Zheng, X. Ning, X. Zeng, and G. Karypis, "Drkg - drug repurposing knowledge graph for covid-19," https://github.com/gnn4dr/DRKG/, 2020.

[75] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 135–142.

[76] T. Jiang, N. D. Sidiropoulos, and J. M. ten Berge, "Almost-sure identifiability of multidimensional harmonic retrieval," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1849–1859, 2001.

[77] X. Jiang, V. Tresp, Y. Huang, and M. Nickel, "Link prediction in multi-relational graphs using additive models." *SeRSy*, vol. 919, pp. 1–12, 2012.

[78] F. Jones, *Lebesgue integration on Euclidean space*. Jones & Bartlett Learning, 2001.

[79] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, and J. C. Ye, "k-t focuss: a general compressed sensing framework for high resolution dynamic mri," *Magnetic resonance in medicine*, vol. 61, no. 1, pp. 103–116, 2009.

[80] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W. Ma, "Hyperspectral super-resolution: A coupled tensor factorization approach," *IEEE Transactions on Signal Processing*, vol. 66, no. 24, pp. 6503–6517, Dec 2018.

[81] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and M. Akçakaya, "Tensor completion from regular sub-nyquist samples," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1–16, 2019.

[82] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and M. Hong, "Structured sumcor multiview canonical correlation analysis for large-scale data," *IEEE Transactions on Signal Processing*, vol. 67, no. 2, pp. 306–319, 2018.

[83] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W.-K. Ma, "Hyperspectral super-resolution: A coupled tensor factorization approach," *arXiv preprint arXiv:1804.05307*, 2018.

[84] ——, "Hyperspectral super-resolution: Combining low rank tensor and matrix structure," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3318–3322.

[85] ——, "Hyperspectral super-resolution via coupled tensor factorization: Identifiability and algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 3191–3195.

[86] C. I. Kanatsoulis and N. D. Sidiropoulos, "Large-scale canonical polyadic decomposition via regular tensor sampling," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[87] ——, "Tex-graph: Coupled tensor-matrix knowledge-graph embedding for covid-19 drug repurposing," 2020.

[88] C. I. Kanatsoulis, N. D. Sidiropoulos, M. Akçakaya, and X. Fu, "Regular sampling of tensor signals: Theory and application to fmri," in *ICASSP 2019-2019 IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 2932–2936.

[89] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 316–324.

[90] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[91] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[92] ——, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[93] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[94] T. G. Kolda, B. W. Bader, and J. P. Kenny, "Higher-order web link analysis using multilinear algebra," in *Proceedings of Fifth IEEE International Conference on Data Mining*. IEEE, 2005, pp. 8–pp.

[95] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[96] V. A. Kotelnikov, "On the transmission capacity of the 'ether'and of cables in electrical communications," in *Proceedings of the first All-Union Conference on the technological reconstruction of the communications sector and the development of low-current engineering. Moscow*. Citeseer, 1933.

[97] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[98] ——, *Multidimensional scaling*. Sage, 1978, no. 11.

[99] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[100] B. Kunkel, F. Blechinger, D. Viehmann, H. V. D. PIEPEN, and R. Doerffer, "Rosis imaging spectrometer and its potential for ocean parameter measurements (airborne and space-borne)," *International Journal of Remote Sensing*, vol. 12, no. 4, pp. 753–761, 1991.

[101] C. Lanaras, E. Baltsavias, and K. Schindler, "Hyperspectral super-resolution by coupled spectral unmixing," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[102] H. Landau, "Necessary density conditions for sampling and interpolation of certain entire functions," *Acta Mathematica*, vol. 117, no. 1, pp. 37–52, 1967.

[103] H. Li, B. Manjunath, and S. K. Mitra, "Multisensor image fusion using the wavelet transform," *Graphical models and image processing*, vol. 57, no. 3, pp. 235–245, 1995.

[104] J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, vol. 3. IEEE, 2008, pp. III–250.

[105] H. Lin, Y. Liu, W. Wang, Y. Yue, and Z. Lin, "Learning entity and relation embeddings for knowledge resolution," *Procedia Computer Science*, vol. 108, pp. 345–354, 2017.

[106] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, "Accelerated dynamic mri exploiting sparsity and low-rank structure: kt slr," *IEEE transactions on medical imaging*, vol. 30, no. 5, pp. 1042–1054, 2011.

[107] J. Liu, "Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details," *International Journal of Remote Sensing*, vol. 21, no. 18, pp. 3461–3472, 2000.

[108] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[109] L. Loncan, L. B. de Almeida, J. M. Bioucas-Dias, X. Briottet, J. Chanussot, N. Dobigeon, S. Fabre, W. Liao, G. A. Licciardi, M. Simoes *et al.*, "Hyperspectral pansharpening: A review," *IEEE Geoscience and remote sensing magazine*, vol. 3, no. 3, pp. 27–46, 2015.

[110] M. Lustig and J. M. Pauly, "SPIRiT: iterative self-consistent parallel imaging reconstruction from arbitrary k-space," *Magnetic resonance in medicine*, vol. 64, no. 2, pp. 457–471, 2010.

[111] C. Ma, B. Clifford, Y. Liu, Y. Gu, F. Lam, X. Yu, and Z.-P. Liang, "High-resolution dynamic 31p-mrsi using a low-rank tensor model," *Magnetic resonance in medicine*, vol. 78, no. 2, pp. 419–428, 2017.

[112] W. K. Ma, J. M. Bioucas-Dias, T. H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C. Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 67–81, Jan 2014.

[113] S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.

[114] M. Mardani, G. B. Giannakis, and K. Ugurbil, "Tracking tensor subspaces with informative random sampling for real-time mr imaging," *arXiv preprint arXiv:1609.04104*, 2016.

[115] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-nyquist sampling of sparse wideband analog signals," *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 375–391, 2010.

[116] S. Moeller, E. Yacoub, C. A. Olman, E. Auerbach, J. Strupp, N. Harel, and K. Uğurbil, "Multiband multislice ge-epi at 7 tesla, with 16-fold acceleration using partial parallel imaging with application to high spatial and temporal whole-brain fmri," *Magnetic resonance in medicine*, vol. 63, no. 5, pp. 1144–1153, 2010.

[117] J. M. Nascimento and J. M. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

[118] M. Newman, *Networks*. Oxford university press, 2018.

[119] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data." in *Icml*, vol. 11, 2011, pp. 809–816.

[120] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.

[121] V. Y. Orekhov, I. Ibraghimov, and M. Billeter, "Optimizing resolution in multidimensional NMR by three-way decomposition," *Journal of biomolecular NMR*, vol. 27, no. 2, pp. 165–173, 2003.

[122] R. Otazo, D. Kim, L. Axel, and D. K. Sodickson, "Combination of compressed sensing and parallel imaging for highly accelerated first-pass cardiac perfusion mri," *Magnetic resonance in medicine*, vol. 64, no. 3, pp. 767–776, 2010.

[123] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1105–1114.

[124] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Parcube: Sparse parallelizable tensor decompositions," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 521–536.

[125] ——, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2017.

[126] E. E. Papalexakis, N. D. Sidiropoulos, and R. Bro, "From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors," *IEEE transactions on signal processing*, vol. 61, no. 2, pp. 493–506, 2013.

[127] K. A. Patwardhan, G. Sapiro, and M. Bertalmío, "Video inpainting under constrained camera motion," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 545–553, 2007.

[128] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[129] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009.

[130] C. Pohl and J. L. Van Genderen, "Review article multisensor image fusion in remote sensing: concepts, methods and applications," *International journal of remote sensing*, vol. 19, no. 5, pp. 823–854, 1998.

[131] H. C. Prescott and T. W. Rice, "Corticosteroids in covid-19 ards: evidence and hope during the pandemic," *Jama*, vol. 324, no. 13, pp. 1292–1295, 2020.

[132] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 459–467.

[133] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[134] S. Rendle and L. Schmidt-Thieme, "Pairwise interaction tensor factorization for personalized tag recommendation," in *Proceedings of the third ACM international conference on Web search and data mining*, 2010, pp. 81–90.

[135] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 74–84.

[136] E. Sanchez and B. R. Kowalski, "Tensorial resolution: a direct trilinear decomposition," *Journal of Chemometrics*, vol. 4, no. 1, pp. 29–45, 1990.

[137] S. S. Schiffman, M. L. Reynolds, and F. W. Young, *Introduction to multidimensional scaling*. Academic press New York, 1981.

[138] M. Selva, B. Aiazzi, F. Butera, L. Chiarantini, and S. Baronti, "Hyper-sharpening: A first approach on sim-ga data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 3008–3024, 2015.

[139] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[140] B. Shaw and T. Jebara, "Structure preserving embedding," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 937–944.

[141] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 3–28, 2003.

[142] P. J. Shin, P. E. Larson, M. A. Ohliger, M. Elad, J. M. Pauly, D. B. Vigneron, and M. Lustig, "Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion," *Magnetic resonance in medicine*, vol. 72, no. 4, pp. 959–970, 2014.

[143] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[144] N. D. Sidiropoulos and G. Z. Dimic, "Blind multiuser detection in w-cdma systems with large delay spread," *IEEE Signal Processing Letters*, vol. 8, no. 3, pp. 87–89, 2001.

[145] N. D. Sidiropoulos, E. E. Papalexakis, and C. Faloutsos, "Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 57–70, 2014.

[146] M. Simões, J. Bioucas-Dias, L. B. Almeida, and J. Chanussot, "A convex formulation for hyperspectral image superresolution via subspace-based regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 6, pp. 3373–3388, 2015.

[147] A. Singhal, "Introducing the knowledge graph: things, not strings," *Official google blog*, vol. 16, 2012.

[148] A. Smilde, R. Bro, and P. Geladi, *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons, 2005.

[149] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis, "Splatt: Efficient and parallel sparse tensor-matrix multiplication," in *2015 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2015, pp. 61–70.

[150] S. M. Smith, C. F. Beckmann, J. Andersson, E. J. Auerbach, J. Bijsterbosch, G. Douaud, E. Duff, D. A. Feinberg, L. Griffanti, M. P. Harms *et al.*, "Resting-state fmri in the human connectome project," *Neuroimage*, vol. 80, pp. 144–168, 2013.

[151] R. Socher, D. Chen, C. D. Manning, and A. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *Advances in neural information processing systems*, 2013, pp. 926–934.

[152] M. Sørensen and L. De Lathauwer, "Fiber sampling approach to canonical polyadic decomposition and tensor completion," *ESAT-STADIUS, KU Leuven, Belgium, Tech. Rep*, pp. 15–151, 2017.

[153] M. Sørensen, I. Domanov, and L. De Lathauwer, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-(lr, n, lr, n, 1) terms—part ii: Algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, pp. 1015–1045, 2015.

[154] M. Sørensen, C. I. Kanatsoulis, and N. D. Sidiropoulos, "Generalized canonical correlation analysis: A subspace intersection approach," *arXiv preprint arXiv:2003.11205*, 2020.

[155] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

[156] Z. Sun, Z.-H. Deng, J.-Y. Nie, and J. Tang, "Rotate: Knowledge graph embedding by relational rotation in complex space," in *International Conference on Learning Representations*, 2018.

[157] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[158] G. Tomasi and R. Bro, "Parafac and missing values," *Chemometrics and Intelligent Laboratory Systems*, vol. 75, no. 2, pp. 163–180, 2005.

[159] W. S. Torgerson, "Multidimensional scaling: I. theory and method," *Psychometrika*, vol. 17, no. 4, pp. 401–419, 1952.

[160] P. A. Traganitis and G. B. Giannakis, "Parafac-based multilinear subspace clustering for tensor data," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2016, pp. 1280–1284.

[161] A. Tsitsulin, D. Mottin, P. Karras, and E. Müller, "Verse: Versatile graph embeddings from similarity measures," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 539–548.

[162] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (aviris)," *Remote sensing of environment*, vol. 44, no. 2-3, pp. 127–143, 1993.

[163] M. A. Veganzones, M. Simoes, G. Licciardi, N. Yokoya, J. M. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 274–288, 2016.

[164] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep Graph Infomax," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=rklz9iAcKQ

[165] R. Venkataramani and Y. Bresler, "Perfect reconstruction formulas and bounds on aliasing error in sub-nyquist nonuniform sampling of multiband signals," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2173–2183, 2000.

[166] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab v3. 0, march 2016," *URL: http://www. tensorlab. net*.

[167] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 284–295, 2016.

[168] N. Vervliet, O. Debals, and L. De Lathauwer, "Canonical polyadic decomposition of incomplete tensors with linearly constrained factors," Technical Report 16–172, ESAT–STADIUS, KU Leuven, Belgium, Tech. Rep., 2017.

[169] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 71–79, 2014.

[170] G. Vivone, L. Alparone, J. Chanussot, M. Dalla Mura, A. Garzelli, G. A. Licciardi, R. Restaino, and L. Wald, "A critical comparison among pansharpening algorithms," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 5, pp. 2565–2586, 2015.

[171] L. Wald, *Data fusion: definitions and architectures: fusion of images of different spatial resolutions*. Presses des MINES, 2002.

[172] L. Wald, T. Ranchin, and M. Mangolini, "Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images," *Photogrammetric Engineering and Remote Sensing*, vol. 63, pp. 691–699, 1997.

[173] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 1225–1234.

[174] Q. Wei, J. Bioucas-Dias, N. Dobigeon, and J.-Y. Tourneret, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3658–3668, 2015.

[175] ——, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3658–3668, 2015.

[176] Q. Wei, J. Bioucas-Dias, N. Dobigeon, J.-Y. Tourneret, M. Chen, and S. Godsill, "Multiband image fusion based on spectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7236–7249, 2016.

[177] Q. Wei, N. Dobigeon, and J.-Y. Tourneret, "Fast fusion of multi-band images based on solving a sylvester equation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4109–4121, 2015.

[178] E. T. Whittaker, "On the functions which are represented by the expansions of the interpolation-theory," *Proceedings of the Royal Society of Edinburgh*, vol. 35, pp. 181–194, 1915.

[179] E. Wycoff, T.-H. Chan, K. Jia, W.-K. Ma, and Y. Ma, "A non-negative sparse promoting algorithm for high resolution hyperspectral imaging," in *Acoustics, Speech and Signal*

Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013, pp. 1409–1413.

[180] B. Yaman, S. Weingärtner, N. Kargas, N. D. Sidiropoulos, and M. Akcakaya, "Locally low-rank tensor regularization for high-resolution quantitative dynamic mri," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2017 IEEE 7th International Workshop on*. IEEE, 2017, pp. 1–5.

[181] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[182] B. Yang, A. Zamzam, and N. D. Sidiropoulos, "Parasketch: Parallel tensor factorization via sketching," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 396–404.

[183] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Chang, "Network representation learning with rich text information," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[184] R. D. Yates and D. J. Goodman, *Probability and Stochastic Processes: S*. John Wiley & Sons, 1998.

[185] N. Yokoya, C. Grohnfeldt, and J. Chanussot, "Hyperspectral and multispectral data fusion: A comparative review of the recent literature," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 2, pp. 29–56, 2017.

[186] N. Yokoya, N. Mayumi, and A. Iwasaki, "Cross-calibration for data fusion of eo-1/hyperion and terra/aster," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 2, pp. 419–426, 2013.

[187] N. Yokoya, T. Yairi, and A. Iwasaki, "Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 528–537, 2012.

[188] M. Yuan and C.-H. Zhang, "On tensor completion via nuclear norm minimization," *Foundations of Computational Mathematics*, vol. 16, no. 4, pp. 1031–1068, 2016.

[189] Y. Zhang and M. He, "Multi-spectral and hyperspectral image fusion using 3-d wavelet transform," *Journal of Electronics (China)*, vol. 24, no. 2, pp. 218–224, 2007.

[190] Z. Zhang and S. Aeron, "Exact tensor completion using t-svd," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2016.

[191] D. Zheng, X. Song, C. Ma, Z. Tan, Z. Ye, J. Dong, H. Xiong, Z. Zhang, and G. Karypis, "Dgl-ke: Training knowledge graph embeddings at scale," *arXiv preprint arXiv:2004.08532*, 2020.

Appendix A

Proofs for Chapter 3

A.1 Proof of Theorem 3.1

First, we note that for any given $\lambda > 0$, the optimal solution to (3.8) should make the two terms zero, when the noise is absent. In other words, our problem boils down to considering if the solution to Problem (3.8) is uniquely determined by $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ up to some trivial ambiguities.

Using Lemma 3.1, it is easily seen that $\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}$ and $\boldsymbol{P}_M\boldsymbol{C}$ are drawn from non-singular absolutely continuous distributions. Therefore, Theorems 2.1 and 2.2 can be employed to characterize the identifiability of the latent factors of the MSI and HSI tensors. Bearing this in mind, recall that the MSI tensor is derived as $\underline{\boldsymbol{Y}}_M = \underline{\boldsymbol{Y}}_S \times_3 \boldsymbol{P}_M$. We note that under Lemma 3.1 and the conditions in the statement of Theorem 3.1, the MSI tensor admits essentially unique latent factors—i.e., if we have $\underline{\boldsymbol{Y}}_M = [\![\boldsymbol{A}_M, \boldsymbol{B}_M, \boldsymbol{C}_M]\!]$, then, the expressions

$$\boldsymbol{A}_M = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1,\ \boldsymbol{B}_M = \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2,\ \boldsymbol{C}_M = \boldsymbol{P}_M\boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3,$$

always hold at the optimality of Problem (3.8), where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}_i$ is a full rank diagonal matrix such that $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$. In other words, by solving (3.8) to optimality, $\boldsymbol{A}$ and $\boldsymbol{B}$ can be identified up to column scaling and permutation ambiguities. To establish identifiablity of $\boldsymbol{C}$, let us consider the HSI tensor, i.e.,

$$\underline{\boldsymbol{Y}}_H = \underline{\boldsymbol{Y}}_S \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2, \tag{A.2}$$

Following the above mdoel, $\underline{\boldsymbol{Y}}_H$ admits a polyadic decomposition (possibly non-unique) $\underline{\boldsymbol{Y}}_H =$

$[\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]$. By matricization, the above can be written as the following:

$$\boldsymbol{Y}_H^{(3)} = (\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{C}^T, \tag{A.3}$$

Plugging in $\boldsymbol{A}_M$ and $\boldsymbol{B}_M$, we have

$$\begin{aligned}\boldsymbol{Y}_H^{(3)} &= (\boldsymbol{P}_2\boldsymbol{B}_M \odot \boldsymbol{P}_1\boldsymbol{A}_M)\boldsymbol{C}^T \\ &= (\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Pi}^2\boldsymbol{C}^T \\ &= (\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{C}_H^T \end{aligned} \tag{A.4}$$

where $\boldsymbol{C}_H = \boldsymbol{C}\boldsymbol{\Lambda}_3\boldsymbol{\Pi}$, which is exactly what we wish to identify. The remaining question is that if $\boldsymbol{C}_H$ can be identified from (C.1)? The answer is affirmative. Indeed, since $\boldsymbol{P}_1\boldsymbol{A}$, $\boldsymbol{P}_2\boldsymbol{B}$ are drawn from absolutely continuous non-singular distributions (cf. Lemma 3.1), we have $\operatorname{krank}(\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A}) = \min\{I_H J_H, F\}$ almost surely [76]. Then, since $I_H J_H \geq F$, the matrix $\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A}$ has full column rank almost surely and $\boldsymbol{C}_H$ can be uniquely identified from (C.1).

We should remark that in the proof we did not use identifiablity of the HSI tensor. This echoes our comment that even the HSI tensor is not identifiable, the super-resolution image can be identified.

A.2 Proof of Theorem 3.2

The proof is simply by applying Theorem 2.1 to the HSI and MSI individually. The reason that we can apply the theorem here is that, by Lemma 3.1, $\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}$ and $\boldsymbol{P}_M\boldsymbol{C}$ are all following some joint absolutely continuous distribution.

Another remark is that although the identifiability can be established by looking at the HSI and MSI individually, the coupled tensor factorization criterion in (3.9) is critical to the reconstruction of the SRI, since the shared parameter $\boldsymbol{C}$ in the two fitting terms serves as an 'anchor' to fix the permutation and scaling ambiguities [cf. Eq (C.1)].

A.3 The spatial degradation model

The proposed work assumes that the forward spatial degradation from SRI to HSI follows the model in (3.4), or equivalently that $\boldsymbol{P}_H$ exhibits a Kronecker structure, i.e. $\boldsymbol{P}_H = \boldsymbol{P}_2 \otimes \boldsymbol{P}_1$.

Here, we prove that the Kronecker structure assumption on $\boldsymbol{P}_H$ is a generalization of the heavily used 2D Gaussian blurring and downsampling procedure, modeled by $\boldsymbol{P}_H\boldsymbol{Y}_S$ in the matricized form [174, 176, 177, 187]. To this end, we show that blurring an image by a Gaussian Kernel and then downsampling is a separable operation across the rows and columns.

Let us assume that $\boldsymbol{\Phi}$ denotes a $q \times q$ Gaussian blurring kernel and $\underline{\boldsymbol{Y}}_S(:,:,k) \in \mathbb{R}^{I_M \times J_M}$ be the matrix representation of the super-resolution image at the kth band. Then the convolution operation of image $\underline{\boldsymbol{Y}}_S(:,:,k)$ with the kernel $\boldsymbol{\Phi}$ can be modeled as:

$$\underline{\boldsymbol{Z}}_H(i,j,k) = \sum_{m=1}^{q}\sum_{n=1}^{q} \boldsymbol{\Phi}(m,n)\underline{\boldsymbol{Y}}_S\left(i-m', j-n', k\right), \tag{A.5}$$

where $m' = m - \lceil \frac{q}{2} \rceil$ and $n' = n - \lceil \frac{q}{2} \rceil$. Here, we have $\boldsymbol{\Phi}(m,n) = (1/2\pi\sigma^2)e^{-\frac{m'^2+n'^2}{2}}$, which can be written as $\boldsymbol{\Phi}(m,n) = \phi(m)\phi(n)$, where $\phi(m) = (1/\sqrt{2\pi\sigma^2})e^{-\frac{m'^2}{2}}$. Then, Eq. (A.5) takes the form

$$\boldsymbol{Z}_H(i,j,k) = \sum_{m=1}^{q}\sum_{n=1}^{q} \phi(m)\phi(n)\underline{\boldsymbol{Y}}_S\left(i-m', j-n', k\right) \tag{A.6}$$

which is a separable 2D convolution operation. Consequently, the blurring processing can be re-written as

$$\underline{\boldsymbol{Z}}_H(:,:,k) = \mathcal{T}_I(\boldsymbol{\phi})\underline{\boldsymbol{Y}}_S(:,:,k)(\mathcal{T}_J(\boldsymbol{\phi}))^T,$$

where $\boldsymbol{\phi} = [\phi(1), \ldots, \phi(q)]^T$ and $\mathcal{T}_l(\boldsymbol{\phi})$ is the Toeplitz matrix that models the 1-D convolution operation of a vector $\boldsymbol{\phi}$ with a vector of size l as a matrix vector multiplication.

The second step of the popular spatial degradation model is to downsample the blurred image by a factor of $d = d_1 d_2$. The 2-D downsampling operation of the blurred image $\boldsymbol{Z}_H$ can be cast as follows:

$$\underline{\boldsymbol{Y}}_H(i,j,k) = \sum_{m=1}^{I}\sum_{n=1}^{J} \boldsymbol{\delta}(m-id_1, n-jd_2)\underline{\boldsymbol{Z}}_H\left(m,n,k\right), \tag{A.7}$$

where $\boldsymbol{\delta}$ is the 2-d Kronecker Delta function. Using the separability property of the 2-D Kronecker Delta (i.e., $\boldsymbol{\delta}(i,j) = \delta(i)\delta(j)$, where $\delta(i)$ is the 1-D Delta function), the transformation from $\underline{\boldsymbol{Y}}_S$ to $\underline{\boldsymbol{Y}}_H$ can be finally modeled as:

$$\underline{\boldsymbol{Y}}_H(:,:,k) = \boldsymbol{S}_1\underline{\boldsymbol{Z}}_H(:,:,k)\boldsymbol{S}_2^T = \boldsymbol{P}_1\underline{\boldsymbol{Y}}_S(:,:,k)\boldsymbol{P}_2^T \tag{A.8}$$

where $\boldsymbol{S}_1, \boldsymbol{S}_2$ are matrices that perform regular sampling of rows and columns respectively (they systematically choose 1 out of d_1 rows and 1 out of d_2 columns of $\underline{\boldsymbol{Z}}_H(:,:,k)$) and $\boldsymbol{P}_1 = \boldsymbol{S}_1\mathcal{T}_I(\boldsymbol{\phi}), \boldsymbol{P}_2 = \boldsymbol{S}_2\mathcal{T}_J(\boldsymbol{\phi})$.

We should mention that although we only showed the Gaussian blurring kernel case here, our tensor mode product based degradation model is compatible with any blurring kernels that factors to row and column blurring operators.

Appendix B

Algorithmic details for Chapter 3

B.1 Initialization algorithms

In this section, we describe the algorithms that we propose to initialize the proposed `STEREO` and `blind STEREO`. The initialization approach computes factors $\boldsymbol{A}$ and $\boldsymbol{B}$ by the rank-F CPD of $\underline{\boldsymbol{Y}}_M$. In the case where the downsampling operator is known, $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are obtained as $\tilde{\boldsymbol{A}} = \boldsymbol{P}_1\boldsymbol{A}$ and $\tilde{\boldsymbol{B}} = \boldsymbol{P}_2\boldsymbol{B}$. Finally factor $\boldsymbol{C}$ is derived as solution to the following linear system of equations:

$$\boldsymbol{Y}_H = (\tilde{\boldsymbol{B}} \odot \tilde{\boldsymbol{A}})\boldsymbol{C}^T \tag{B.1}$$

The initialization algorithm, named as *Tensor Reconstruction* (`TenRec`) is given in Algorithm 1.

Algorithm B.1 `TenRec`

Initialization: F
$\boldsymbol{A}, \boldsymbol{B}, \tilde{\boldsymbol{C}} \leftarrow \text{CPD}(\underline{\boldsymbol{Y}}_M)$
$\tilde{\boldsymbol{A}} \leftarrow \mathbf{P}_1\boldsymbol{A}$
$\tilde{\boldsymbol{B}} \leftarrow \mathbf{P}_2\boldsymbol{B}$
$\boldsymbol{C} \leftarrow$ solve (B.1)

In case where $\boldsymbol{P}_H$ is unknown, $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{B}}$ are approximated by averaging out $d = \frac{I_M}{I_H}$ column entries of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively, to roughly mimic the blurring and downsampling process.

Then matrix $\boldsymbol{C}$ can then be obtained as before. Algorithm B.2 describes the algorithm.

Algorithm B.2 `Blind TenRec`

Initialization: F
$\boldsymbol{A}, \boldsymbol{B}, \tilde{\boldsymbol{C}} \leftarrow \text{CPD}(\underline{\boldsymbol{Y}}_M)$
$\tilde{\boldsymbol{A}}(i,:) \leftarrow \sum_{k=d(i-1)+1}^{di} \boldsymbol{A}(k,:)$
$\tilde{\boldsymbol{B}}(i,:) \leftarrow \sum_{k=d(i-1)+1}^{di} \boldsymbol{B}(k,:)$
$\boldsymbol{C} \leftarrow$ solve (B.1)

B.2 Sylvester solution to `STEREO` subproblems

We discuss the `STEREO` updates to variables $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$. To make this argument concrete, take for example the update for $\boldsymbol{A}$ in Algorithm 1:

$$\boldsymbol{A} \leftarrow \arg\min_{\boldsymbol{A}} \|\boldsymbol{Y}_H^{(1)} - (\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})\boldsymbol{A}^T\boldsymbol{P}_1^T\|_F^2 + \lambda\|\boldsymbol{Y}_M^{(1)} - (\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{A}^T\|_F^2. \tag{B.2}$$

Taking the derivative and setting it equal to $\mathbf{0}$ yields the following Sylvester equation:

$$\begin{aligned}&\lambda\boldsymbol{A}(\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{B})^T(\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{B}) + \boldsymbol{P}_1^T\boldsymbol{P}_1\boldsymbol{A}(\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})^T(\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})\\ &= \lambda\boldsymbol{Y}_M^{(1)^T}(\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{B}) + \boldsymbol{P}_1^T\boldsymbol{Y}_H^{(1)^T}(\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B}).\end{aligned} \tag{B.3}$$

Vectorizing (B.3), will give a least square update for $\boldsymbol{A}$:

$$\boldsymbol{Q}\text{vec}(\boldsymbol{A}) = \text{vec}(\lambda\boldsymbol{Y}_M^{(1)^T}(\boldsymbol{P}_M\boldsymbol{C} \odot \boldsymbol{B}) + \boldsymbol{P}_1^T\boldsymbol{Y}_H^{(1)^T}(\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})), \tag{B.4}$$

where $\boldsymbol{Q} = \lambda(\boldsymbol{P}_M\boldsymbol{C}\odot\boldsymbol{B})^T(\boldsymbol{P}_M\boldsymbol{C}\odot\boldsymbol{B})\otimes\boldsymbol{I} + (\boldsymbol{C}\odot\boldsymbol{P}_2\boldsymbol{B})^T(\boldsymbol{C}\odot\boldsymbol{P}_2\boldsymbol{B})\otimes\boldsymbol{P}_1^T\boldsymbol{P}_1 \in \mathbb{R}^{I_MF\times I_MF}$. The complexity for solving (B.4) is $\mathcal{O}(I_M^3F^3)$, which can be prohibitive for large I_M or F. In addition, storing $\boldsymbol{Q}$ in a naive way can be challenging. Hence, instead of solving (B.4), we choose to approach (B.3) by utilizing efficient numerical algorithms for solving the Sylvester equation which need $\mathcal{O}(I_M^3)$ flops and are less memory demanding [20, 58].

Appendix C

Proofs for Chapter 4

C.1 Proof of Theorem 4.1

First, we adjust Lemma 3.1 from Chapter 3 (Lemma 1 in [80]) for complex numbers and selection matrices, which is essential for the proofs.

Lemma C.1. *Let $\tilde{\boldsymbol{Z}} = \boldsymbol{Q}\boldsymbol{Z}$, where the elements of $\boldsymbol{Z}$ are drawn from an absolutely continuous joint distribution with respect to the Lebesgue measure in $\mathbb{F}^{IF}$ and $\boldsymbol{Q} \in \mathbb{R}^{I' \times I}$ is a row selection matrix with full row rank. Then the joint distribution of the elements in $\tilde{\boldsymbol{Z}}$ is absolutely continuous with respect to the Lebesgue measure in $\mathbb{F}^{I'F}$*

It follows that $\boldsymbol{P}_1^{(1)}\boldsymbol{A}, \boldsymbol{P}_3^{(2)}\boldsymbol{C}$ are drawn from non-singular absolutely continuous joint distributions. Then Theorem 2.1 determines the conditions under which the factors of $\underline{\boldsymbol{Y}}_1$ or $\underline{\boldsymbol{Y}}_2$ can be identified. Let's consider the case where $\underline{\boldsymbol{Y}}_1$ is identifiable. Under the conditions of Theorem 4.1, $\underline{\boldsymbol{Y}}_1 = [\![\boldsymbol{A}_1, \boldsymbol{B}_1, \boldsymbol{C}_1]\!]$ is essentially unique and from (4.1) holds that:

$$\boldsymbol{A}_1 = \boldsymbol{P}_1^{(1)}\boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1,\ \boldsymbol{B}_1 = \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2,\ \boldsymbol{C}_1 = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3,$$

where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}_i$ is a full rank diagonal matrix such that $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$. Recall that $\underline{\boldsymbol{Y}}_2$ admits a (possibly non-unique) PD $\underline{\boldsymbol{Y}}_2 = \left[\!\!\left[\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3^{(2)}\boldsymbol{C}\right]\!\!\right]$. Matricizing $\underline{\boldsymbol{Y}}_2$ and plugging in $\boldsymbol{B}_1$ and $\boldsymbol{C}_1$ leads to:

$$\boldsymbol{Y}_2^{(1)} = (\boldsymbol{P}_3^{(2)}\boldsymbol{C}_1 \odot \boldsymbol{B}_1)\boldsymbol{A}_2^T = \tag{C.1a}$$

$$(\boldsymbol{P}_3^{(2)}\boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3 \odot \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2)\boldsymbol{A}_2^T = \tag{C.1b}$$

$$(\boldsymbol{P}_3^{(2)}\boldsymbol{C}\boldsymbol{\Lambda}_3'\boldsymbol{\Pi} \odot \boldsymbol{B}\boldsymbol{\Lambda}_2'\boldsymbol{\Pi})\boldsymbol{A}_2^T = \tag{C.1c}$$

$$(\boldsymbol{P}_3^{(2)}\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{\Lambda}_2'\boldsymbol{\Lambda}_3'\boldsymbol{\Pi}\boldsymbol{A}_2^T = \tag{C.1d}$$

$$(\boldsymbol{P}_3^{(2)}\boldsymbol{C} \odot \boldsymbol{B})\boldsymbol{\Pi}\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3\boldsymbol{A}_2^T \tag{C.1e}$$

In equation (C.1c), (C.1e) we have used the property that $\boldsymbol{\Pi}\boldsymbol{\Lambda}_3 = \boldsymbol{\Lambda}_3'\boldsymbol{\Pi}$, where $\boldsymbol{\Lambda}_3'$ is a diagonal matrix whose diagonal entries have been permuted according to $\boldsymbol{\Pi}$ and equation (C.1e) is due to the definition of the Khatri-Rao product. Since $JK_2 \geq F$, then $(\boldsymbol{P}_3^{(2)}\boldsymbol{C}) \odot \boldsymbol{B}$ has full column rank almost surely [76], and $\boldsymbol{A}_2 = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1$ can be identified from (C.1). Therefore $\hat{\underline{\boldsymbol{X}}} = [\![\boldsymbol{A}_2, \boldsymbol{B}_1, \boldsymbol{C}_1]\!]$ reconstructs signal $\underline{\boldsymbol{X}}$.

C.2 Proof of Theorems 4.2, 4.3

To begin, we use Lemma C.1 and observe that $\boldsymbol{P}_1^{(d)}\boldsymbol{A}$, $\boldsymbol{P}_2^{(d)}\boldsymbol{B}, \boldsymbol{P}_3^{(d)}\boldsymbol{C}$ are drawn from non-singular absolutely continuous distributions. Note that $\boldsymbol{P}_1^{(d)}, \boldsymbol{P}_2^{(d)}, \boldsymbol{P}_3^{(d)}$ have full row rank by construction. Then we use Theorem 2.1 to claim identifiability of the factors of each sub-tensor $\underline{\boldsymbol{Y}}_d$. Under the conditions of Theorems 4.2, 4.3, $\boldsymbol{P}_1^{(d)}\boldsymbol{A}, \boldsymbol{P}_2^{(d)}\boldsymbol{B}, \boldsymbol{P}_3^{(d)}\boldsymbol{C}$ can be identified, which corresponds to identifying all the rows of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$, up to column permutation and scaling. The caveat is that the rows of the factors are subject to column permutation and scaling mismatch, since they are obtained by the CPD of independent sub-sampled tensors. For example, let $\underline{\boldsymbol{Y}}_d = [\![\boldsymbol{A}_d, \boldsymbol{B}_d, \boldsymbol{C}_d]\!]$ and $\underline{\boldsymbol{Y}}_{d'} = [\![\boldsymbol{A}_{d'}, \boldsymbol{B}_{d'}, \boldsymbol{C}_{d'}]\!]$. Then, from equations (4.9), it becomes clear that in order to obtain $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$ from $\boldsymbol{A}_d, \boldsymbol{B}_d, \boldsymbol{C}_d,\ d = 1, \ldots, D$ and complete $\underline{\boldsymbol{X}}$, the permutation and scaling mismatch should be resolved, i.e., $\boldsymbol{\Pi}^{(d)} = \boldsymbol{\Pi}^{(d')}$, $\boldsymbol{\Lambda}_i^{(d)} = \boldsymbol{\Lambda}_i^{(d')}$ for every $d,\ d'$. To do so the following lemma is being used:

Lemma C.2. *Assume the entries of $\boldsymbol{C} \in \mathbb{F}^{K \times F}$ are jointly drawn from an absolutely continuous distribution over $\mathbb{F}^{KF}$. Then $\boldsymbol{C}(i, f) \neq \boldsymbol{C}(i', f')$ almost surely.*

Proof. The proof is similar to the proof of Corollary 1 in [76] and uses the fact that $\boldsymbol{C}(i, f) - \boldsymbol{C}(i', f')$ is a non-trivial analytic function of the entries of $\boldsymbol{C}$ and thus $\boldsymbol{C}(i, f) - \boldsymbol{C}(i', f') \neq 0$ almost surely. □

Finally, to resolve the mismatch, we utilize the rules in (4.4), (4.6). Particularly, when the original tensor is fiber sampled $\boldsymbol{C}_d = \boldsymbol{C}_{d'} = \boldsymbol{C},\ \forall d,\ d'$ up to column permutation and scaling. Then, column permutation can be fixed to be the same for all $\underline{\boldsymbol{Y}}_d$s, since the entries of $\boldsymbol{C}$ are

not equal almost surely. In order to reconcile for scaling mismatch, (4.4c), guarantees that there exist at least one row of $\boldsymbol{A}$ or $\boldsymbol{B}$ that is identified (up to permutation and scaling) from 2 different sub-sampled tensors $\underline{\boldsymbol{Y}}_d$. This is sufficient to resolve the CPD scaling mismatch between every $\underline{\boldsymbol{Y}}_d$-$\underline{\boldsymbol{Y}}_{d'}$ couple, due to Lemma C.2. The entry sampling mechanism, differs to the fiber sampling one, in the fact that $\boldsymbol{C}$ can only be partially identified from each sub-sampled version $\underline{\boldsymbol{Y}}_d$. Following same principles as before, permutation and scaling mismatch on the CPD of different $\underline{\boldsymbol{Y}}_d$s is resolved by (4.6d) along with Lemma C.2.

C.3 Proof of Theorems 5.2, 4.5

The proof is similar to that of Theorem 4.1, 4.2, 4.3. The main difference lies in the fact that Theorem 2.3 is now employed, to establish identifiability on the CPD of each sub-sampled tensor and therefore recoverability of the original tensor. Furthermore, permutation and scaling alignment is performed using the rows of the latent factors which are common among the sub-tensors. This is accomplished, since factors with repeated entries are not allowed.

Appendix D

Algorithmic details for Chapter 6

We are given an adjacency matrix $\boldsymbol{Y}^1 = \boldsymbol{S}_{\mathcal{G}} \in \{0,1\}^{N \times N}$ and a matrix of node attributes $\boldsymbol{Y}^2 = \mathcal{A} \in \mathbb{R}^{N \times d}$. We are interested in computing the CPD of tensor $\underline{\boldsymbol{X}}$ with:

$$\underline{\boldsymbol{X}}(:,:,1) = \boldsymbol{X}^1 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^1\boldsymbol{Y}^{1^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right), \tag{D.1}$$

$$\underline{\boldsymbol{X}}(:,:,2) = \boldsymbol{X}^2 = \left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}\left(\boldsymbol{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right) \tag{D.2}$$

The objective of this appendix is to show how to perform this CPD computation by exploiting the special sparsity structure and without instantiating a dense $\underline{\boldsymbol{X}}$.

D.1 Efficient CPD computations for GAGE-EVD

The first step of GAGE algorithm involves an eigenvalue decomposition approach. The bottleneck operation is:

$$\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T \leftarrow \text{EVD}\left(\boldsymbol{X}^{(1)^T}\boldsymbol{X}^{(1)}, F\right) \tag{D.3}$$

First let us observe the structure of matrix $\boldsymbol{X}^{(1)^T}\boldsymbol{X}^{(1)}$. Note that $\boldsymbol{X}^{(1)} = \begin{bmatrix}\boldsymbol{X}^1 \\ \boldsymbol{X}^2\end{bmatrix}$, and $\boldsymbol{X}^1,\ \boldsymbol{X}^2$ are both symmetric matrices.

$$\boldsymbol{X}^{(1)^T}\boldsymbol{X}^{(1)} = \begin{bmatrix}\boldsymbol{X}^{1^T} \boldsymbol{X}^{2^T}\end{bmatrix}\begin{bmatrix}\boldsymbol{X}^1 \\ \boldsymbol{X}^2\end{bmatrix} = \boldsymbol{X}^{1^T}\boldsymbol{X}^1 + \boldsymbol{X}^{2^T}\boldsymbol{X}^2 \tag{D.4}$$

$$\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}_1\boldsymbol{Y}^{1^T}\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^1\boldsymbol{Y}^{1^T}\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)+ \tag{D.5}$$

$$\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{Y}^2\boldsymbol{Y}^{2^T}\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right), \tag{D.6}$$

since $\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)=\left(\boldsymbol{I}-\frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$. To compute the EVD of $\boldsymbol{X}^{(1)^T}\boldsymbol{X}^{(1)}$ we resort to the orthogonal iterations method [59]. The steps are summarized as follows:

- Initialize $\boldsymbol{Q}_0 \in \mathbb{R}^{N\times F}$: orthogonal matrix
  repeat:
- $\boldsymbol{W}_k = \boldsymbol{X}^{1^T}\boldsymbol{X}^1\boldsymbol{Q}_{k-1} + \boldsymbol{X}^{2^T}\boldsymbol{X}^2\boldsymbol{Q}_{k-1}$
- $\boldsymbol{Q}_k \leftarrow \text{QR}\left(\boldsymbol{W}_k\right)$
  until convergence

In the first step of the loop every computation is either a sparse or rank 1 multiplication which can be performed efficiently. The computationally more intensive computation lies in the QR computation of matrix $\boldsymbol{W}_k$. The complexity of this step is $\mathcal{O}(NF^2)$ which is linear in the number of nodes.

D.2 Sparsity aware GAGE

Now we study the ALS updates in GAGE algorithm. The update for $\boldsymbol{U}$ can be written as:

$$\boldsymbol{U} \leftarrow \text{solve}\left(\left(\boldsymbol{C}^T\boldsymbol{C}\right) * \left(\boldsymbol{U}'^T\boldsymbol{U}'\right)\right)\boldsymbol{U}^T = \left(\boldsymbol{C}\odot\boldsymbol{U}'\right)^T\boldsymbol{X}^{(1)}. \tag{D.7}$$

The matrix matrix multiplication in the right hand side can exploit the special structure of $\boldsymbol{X}^{(1)}$:

$$\left(\boldsymbol{C}\odot\boldsymbol{U}'\right)^T\boldsymbol{X}^{(1)} = \begin{bmatrix}\boldsymbol{U}'\text{diag}\left(\boldsymbol{C}(1,:)\right)\\ \boldsymbol{U}'\text{diag}\left(\boldsymbol{C}(2,:)\right)\end{bmatrix}^T\begin{bmatrix}\boldsymbol{X}^1\\ \boldsymbol{X}^2\end{bmatrix} =$$

$$\sum_{k=1}^{2}\text{diag}\left(\boldsymbol{C}(k,:)\right)\boldsymbol{U}'^T\boldsymbol{X}_k. \tag{D.8}$$

Leveraging (6.7), it follows that the number of flops required to compute (D.8) is $\mathcal{O}(sF)$, where $s = s_1 + s_2$ and $s_1,\ s_2$ are the number of non-zeros in $\boldsymbol{Y}^1, \boldsymbol{Y}^2$ respectively. The same principles

hold for the update of $\boldsymbol{U}^{'}$:

$$\boldsymbol{U}^{'} \leftarrow \text{solve}\left(\left(\boldsymbol{C}^T\boldsymbol{C}\right) * \left(\boldsymbol{U}^T\boldsymbol{U}\right)\right)\boldsymbol{U}^{'T} = \left(\boldsymbol{C} \odot \boldsymbol{U}\right)^T \boldsymbol{X}^{(2)}. \tag{D.9}$$

$$\left(\boldsymbol{C} \odot \boldsymbol{U}\right)^T \boldsymbol{X}^{(2)} = \begin{bmatrix}\boldsymbol{U}\text{diag}\left(\boldsymbol{C}(1,:)\right)\\ \boldsymbol{U}\text{diag}\left(\boldsymbol{C}(2,:)\right)\end{bmatrix}^T \begin{bmatrix}\boldsymbol{X}^1\\ \boldsymbol{X}^2\end{bmatrix} =$$
$$\sum_{k=1}^{2} \text{diag}\left(\boldsymbol{C}(k,:)\right)\boldsymbol{U}^T\boldsymbol{X}_k. \tag{D.10}$$

The update of $\boldsymbol{C}$ can be written as:

$$\boldsymbol{C} \leftarrow \text{solve}\left(\left(\boldsymbol{U}^{'T}\boldsymbol{U}^{'}\right) * \left(\boldsymbol{U}^T\boldsymbol{U}\right)\right)\boldsymbol{C}^T = \left(\boldsymbol{U}^{'} \odot \boldsymbol{U}\right)^T \boldsymbol{X}^{(3)}. \tag{D.11}$$

To avoid instantiating $\boldsymbol{U}^{'} \odot \boldsymbol{U}$, we observe that:

$$\left(\boldsymbol{U}^{'} \odot \boldsymbol{U}\right)^T \boldsymbol{X}^{(3)} = \begin{bmatrix}\boldsymbol{U}(:,1)^T\boldsymbol{X}^1U^{'}(:,1), \boldsymbol{U}(:,1)^T\boldsymbol{X}^2U^{'}(:,1)\\ \boldsymbol{U}(:,2)^T\boldsymbol{X}^1U^{'}(:,2), \boldsymbol{U}(:,2)^T\boldsymbol{X}^2U^{'}(:,2)\\ \vdots\\ \boldsymbol{U}(:,F)^T\boldsymbol{X}^1U^{'}(:,F), \boldsymbol{U}(:,F)^T\boldsymbol{X}^2U^{'}(:,F)\end{bmatrix} \tag{D.12}$$

The operation in (D.12) avoids storing $\boldsymbol{U}^{'} \odot \boldsymbol{U}$ and can also exploit the structure in $\boldsymbol{X}^1,\ \boldsymbol{X}^2$. The overall operation can be computed efficiently in $\mathcal{O}(max\{NF, sF\})$ flops.

Appendix E

Algorithmic details for Chapter 7

`TeX-Graph` solves the following problem

$$\text{minimize}_{\{\boldsymbol{A}_m\},\{\boldsymbol{C}_{m,n}\}} \sum_{(m,n)\in\mathcal{S}} \left\| \underline{\boldsymbol{X}}_{m,n} - [\![\boldsymbol{A}_m, \boldsymbol{A}_n, \boldsymbol{C}_{m,n}]\!] \right\|_F^2 . \tag{E.1}$$

Then the update for $\boldsymbol{A}_n$ is the solution of:

$$\text{minimize}_{\boldsymbol{A}_n} \sum_{m\in\mathcal{S}_n^+} \left\| \underline{\boldsymbol{X}}_{m,n} - [\![\boldsymbol{A}_m, \boldsymbol{A}_n, \boldsymbol{C}_{m,n}]\!] \right\|_F^2 + \sum_{p\in\mathcal{S}_n^-} \left\| \underline{\boldsymbol{X}}_{n,p} - [\![\boldsymbol{A}_n, \boldsymbol{A}_p, \boldsymbol{C}_{n,p}]\!] \right\|_F^2 , \tag{E.2}$$

where $\mathcal{S}_n^+, \mathcal{S}_n^-$ are defined in (7.9). Problem (E.2) can be written as:

$$\text{minimize}_{\boldsymbol{A}_n} \sum_{m\in\mathcal{S}_n^+} \left\| \boldsymbol{X}_{m,n}^{(1)} - (\boldsymbol{C}_{m,n} \odot \boldsymbol{A}_m) \, \boldsymbol{A}_n{}^T \right\|_F^2 + \sum_{p\in\mathcal{S}_n^-} \left\| \boldsymbol{X}_{n,p}^{(2)} - (\boldsymbol{C}_{n,p} \odot \boldsymbol{A}_p) \, \boldsymbol{A}_n{}^T \right\|_F^2 . \tag{E.3}$$

Taking the gradient of (E.3) with respect to $\boldsymbol{A}_n$ and setting it to zero yields the equation in (7.8). The main bottleneck of (7.8) in terms of memory requirements and computational complexity is instantiating the Khatri-Rao products $(\boldsymbol{C}_{n,p} \odot \boldsymbol{A}_p)$, $(\boldsymbol{C}_{m,n} \odot \boldsymbol{A}_m)$ and computing the MTTKRP $(\boldsymbol{C}_{n,p} \odot \boldsymbol{A}_p)^T \boldsymbol{X}_{n,p}^{(1)}$, $(\boldsymbol{C}_{m,n} \odot \boldsymbol{A}_m)^T \boldsymbol{X}_{m,n}^{(2)}$. We focus on the computation of:

$$(\boldsymbol{C}_{n,p} \odot \boldsymbol{A}_p)^T \boldsymbol{X}_{n,p}^{(1)}. \tag{E.4}$$

Equation (E.4) can be equivalently written as:

$$\begin{bmatrix} \boldsymbol{A}_p \text{diag}\left(\boldsymbol{C}_{n,p}(1,:)\right) \\ \vdots \\ \boldsymbol{A}_p \text{diag}\left(\boldsymbol{C}_{n,p}(K_{n,p},:)\right) \end{bmatrix}^T \begin{bmatrix} \boldsymbol{X}_{n,p}^{1^T} \\ \vdots \\ \boldsymbol{X}_{n,p}^{K_{n,p}^T} \end{bmatrix} = \sum_{k=1}^{K_{n,p}} \text{diag}\left(\boldsymbol{C}_{n,p}(k,:)\right) \boldsymbol{A}_p \boldsymbol{X}_{n,p}^{k^T}. \tag{E.5}$$

It is clear from equation (E.5) that $(\boldsymbol{C}_{n,p} \odot \boldsymbol{A}_p)$ need not be instantiated. Furthermore, the number of flops to compute (E.5) is $\mathcal{O}(F \cdot \text{nnz}(\underline{\boldsymbol{X}}_{n,p}))$. Note that computing $\left(\boldsymbol{C}_{m,n} \odot \boldsymbol{A}_m\right)^T \boldsymbol{X}_{m,n}^{(2)}$ is only different in the fact that the frontal slabs are not transposed, and is thus omitted.

The update for $\boldsymbol{C}_{m,n}$ is the solution of:

$$\text{minimize}_{\boldsymbol{C}_{m,n}} \ \left\| \underline{\boldsymbol{X}}_{m,n} - [\![\boldsymbol{A}_m, \boldsymbol{A}_n, \boldsymbol{C}_{m,n}]\!] \right\|_F^2, \tag{E.6}$$

or equivalently:

$$\text{minimize}_{\boldsymbol{C}_{m,n}} \ \left\| \boldsymbol{X}_{m,n}^{(3)} - \left(\boldsymbol{A}_m \odot \boldsymbol{A}_n\right) {\boldsymbol{C}_{m,n}}^T \right\|_F^2. \tag{E.7}$$

Taking the gradient of (E.7) with respect to $\boldsymbol{C}_{m,n}$ and setting it to zero yields the equation in (7.10). The main memory and computation bottleneck of equation (7.10) is computing the MTTKRP. The formula in (E.5) can be utilized if $\boldsymbol{C}_{n,p}$ is replaced by $\boldsymbol{A}_n$, $\boldsymbol{A}_p$ is replaced by $\boldsymbol{A}_m$ and the transposed frontal slabs $\boldsymbol{X}_{m,n}^{k^T}$ are replaced by vertical slabs.